AN ARGUS SPECIALIST PUBLICATION

Dec 1985/Jan 1986

£1.95

ZX COMPUTING

The Magazine For All Sinclair Users

CAD EQUIPMENT REVIEW



MIDI SOFTWARE OFFER

SCREENMASTER BRUSH UP ON YOUR GRAPHICS

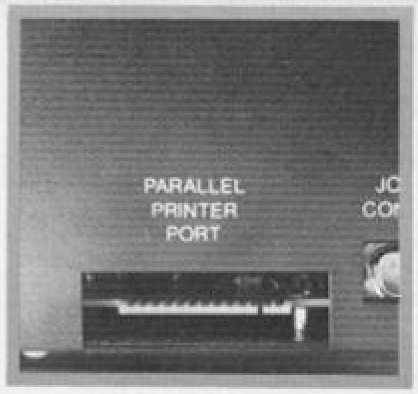
QL MATCHPOINT COMPETITION

With More Editorial Pages Than Any Monthly Magazine


HANTAREX
Opus.


Opus.

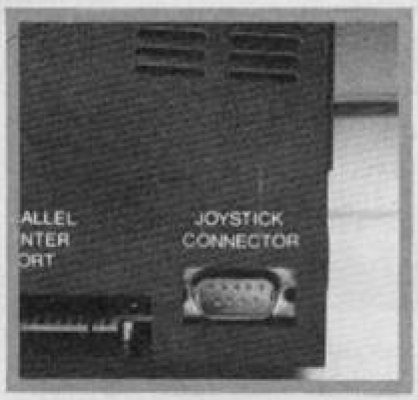

PERIPHERAL
THROUGH
CONNECTOR


PARALLEL
PRINTER
PORT


VIDEO
CONNECTOR

TECHNICAL DETAILS
3.5" 250 K disc drive Double density disc interface Parallel printer interface Joystick interface Video monitor interface Peripheral through connector Built-in power supply Utilities on ROM including format and verify RAM disc facility Random access files fully supported Connections for second drive upgrade Comprehensive user manual
Total: £199.95 inc. VAT







"Recommended" – Sinclair User Buyers' Guide

Discovery – helps your Spectrum make all the best connections for only £199.95

In one simple step you can now transform your Spectrum into a complete computer system. Discovery from Opus Supplies is a sophisticated computer nerve centre combining not just a powerful disc drive and double density disc interface but the most comprehensive range of interfaces capable of running printers, monitors, joysticks and other peripherals.

No other system can offer your Spectrum so many options – allowing you to play the very latest games or providing facilities for serious business use. Discovery's specification includes over £300 worth of Spectrum peripherals and yet, at just £199.95, this remarkable all-in-one system costs less than a disc drive and interface combination.

Just connect Discovery to the rear of your Spectrum and you've started the transformation. Now you have immediate access to the very latest Japanese 3½" disc drive controlled via a powerful and ingenious double density disc interface. The unit's built-in power supply feeds both Discovery and your computer allowing you to dispense with your Sinclair power supply.

The Opus system takes no user memory from your computer giving you full compatability with all tape-based software including programs designed for use with microdrive and Discovery even has its own RAM disc allowing you super fast access times.

Now the choice is yours. Connect up a joystick or a printer to the built-in Discovery interfaces, link-up to a video monitor or incorporate your own choice of peripheral using the through connector.

Whether you choose Discovery for games or business use you can rely on a complete range of readily available disc software produced by many of Britain's leading software companies, and our price of £199.95 includes an unbeatable two-year free warranty and free delivery.

Discovery is available nationwide from Boots and from all good computer stores across the country. Ring us now on 0737-65080 for details of your nearest dealer, or to order your Discovery.

Trade and export enquiries welcome.

*Price quoted refers to Discovery unit only.

Opus. Opus Supplies Ltd

Opus Supplies Ltd,
55 Ormside Way,
Holmethorpe Industrial Estate,
Redhill, SURREY RH1 2LW.

Editor: Ray Elder
Editorial Assistant: Cliff Joseph
Group Editor: Wendy J Palmer
Software Assistant: John Gerard Donovan
Sales Executive: Alice Robertson
Advertisement Manager: Barry Bingham
Divisional Advertising Manager: Chris Northam
Production Controller: Sue Couchman
Publishing Director: Peter Welham

Origination and design by Argus Design Ltd., Circus House, 26 Little Portland Street, London W1N 5AF.

Published by Argus Specialist Publications Ltd, 1 Golden Square, London W1R 3AB.

CONTENTS

ASP FIGHTS SOFTWARE PIRACY

Much has been said and written in condemnation of software piracy but few have taken a positive stand against it. ASP is among those few that have taken action to help curb the grave problem of home copying of commercial software.

ASP has already taken steps to eliminate advertisements in our magazines which relate to tape duplication for piracy purposes. While it is appreciated that individuals may take 'back-up' copies of their own programs, it should be noted that it is *ILLEGAL* to copy commercially available software for other than personal use.

Software piracy is costing the software industry huge sums of money which is detrimental to the future development of the industry. It is in everybody's interests to dramatically reduce the level of software piracy primarily because firms need funds raised from software sales to plough back into research and development of new products. This means that the standard of software products can only improve.

ASP hopes our action will help combat this serious problem in order to maintain and improve the high standards of the UK software industry. We are asking *you* to do the same by refraining from duplicating or copying commercially available software for anything other than personal use.





MPUTING

ZX Computing is published bi-monthly on the fourth Friday of the month. Distributed by: Argus Press Sales & Distribution Ltd, 12-18 Paul Street, London EC2A 4JS. 01-247 8233. Printed in the UK by: Garnett Print, Rotherham and London.







At last as we all knew would happen, Sinclair Research lowered the price of the QL in September. Of course this is old news by now, but the effect will only just be starting to be seen and felt.

The QL was always a superb machine, at first in specifications only but latterly, as the QDOS was improved and corrected, in reality. SuperBASIC is aptly named, I prefer it to the Beeb's much vaunted BASIC, and the possibilities of the 68000 processor have still to be completely exploited. The biggest problem was (and still is in my experience) the microdrives. Even with a computer new in August, after making only six backups from Quill (for the purposes of reviewing disk drives) the original has been ruined.



Disk drives are the answer of course, and several interfaces have been on the market for some time now. However the combined cost has been prohibitive — in the realms of £650+ for the smallest system. By knocking £200 off this total the QL becomes not only a viable proposition but, taking the Psion software into consideration, a very good deal indeed for the serious/small business user. Just as importantly the basic machine now at £199.95 is within the price limits of many others, hobbyists, games players and those wishing to learn about computers. Remember the Spectrum in its original rubber key form was only £25 cheaper when first launched and it sold like hot cakes!

Yes, I know that the incredible development of technology has lowered cost and increased the ability of computers since then, but even so the QL now represents one of the best value for money deals around. So more people will buy the machine, therefore the software market will (hopefully) expand and programs will become more available, there is quite a reasonable amount available already (though the qualty of some is rather dubious), the cost of software should come down and hopefully the QL should take off.

And so to bed

This issue contains a special feature on graphics, an area which most of us use in one way or another. From the simplest games to the most complex business program, all benefit from the judicious use of graphics. We have attempted to cover a wide area in this feature and we are sure there is something here for you, whatever your particular interest.

And, last but certainly not least, may we all wish you a very happy ZXmas and one QL of a New Year!

Ray and Cliff

Cribbage

Dear ZX Computing,
The Cribbage program in your Aug/Sept issue is great — the graphics are good (although the pegs don't quite get it right) and the game strategy employed by the Spectrum has quite often frustrated my own! However, the program as it stands is not quite the same as the one I grew up with.

1. You must 'peg into the hole', ie score exactly 121 points.
2. You cannot go out on a Jack! (I mean when you're on 119 and the other cuts you a Jack).

I cured this with the following lines:

```
2925 IF s(player)+s 121
  THEN LET s=0
6002 same text as 2925
```

The following amendment is also required:

```
8020 IF k=11 THEN. . .
```

must be amended to

```
8020 IF k=11 AND s(player)
  119 THEN. . .
```

I believe that I have sufficiently tested my version of this program (15-20 times) to be able to say that, not only does it work but, due to the Spectrum's penchant for flushes, it can also find itself the occasional 'pegging hand' when the crunch comes. As the Spectrum has won a few more games than I have, the program doesn't seem to lose any quality by introducing these rules. The text of the instructions must also be amended to clarify these adjustments (line 9085 PRINT "to reach quality" "121 points" "(twice round the board)." and an additional line 9550 PRINT AT 5,0;"A card is now" "cut from the pack." "If a jack is cut" "from the pack then" "the dealer gets" "2 points, unless" "he already has" "119 points (you" "cannot go out on" "a jack)." :PAUSE 0:GO SUB clear).

I would also like to thank you and A.J. Carter for the article 'Adding the Plus' from the same issue. By pure chance I had ordered an upgrading kit a few days before buying your publication, and despite now having to wait for the kit to arrive (understandable, due to the summer holidays of the staff at the mail order firm) I have 'upgraded' my Spectrum about ten times by now and can't wait for the kit to arrive so that I can get on with the real thing! The article is interesting, helps to reduce the frustration of waiting and may well prove useful when the upgrading kit arrives.
Yours sincerely,
Philip Baker
West Germany

What does it do II?

Dear Sir,
I have just read the article 'What does it do?' by John Ure. I have followed much the same course, but without any of the many problems he seems to have had.

I started by buying Interface 1 and Microdrive — no problem with fitting this unit, it all went together in a few minutes and worked first time. I then bought Tasword 2 and ten minutes' work amending the program according to Tasman instructions saw it saved on a Microdrive cartridge. At this point, it was clear that I needed a decent keyboard, so I opted for the expensive Transform unit. Delivery was prompt, ten days and fitting was easy, about fifteen minutes. The next step was a printer, so once again I went to Transform and bought the Smith-Corona TP1. I asked Transform to send it to my work address, and they actually rang me up at work to check that I had the RS232 connector — I had not given my work phone number! The printer arrived two days later, ten days after it was ordered. The Sinclair RS232 connector came into use and after amending Tasword 2 to include the baud rate and saving the amended version, I was in business. The only problem I had was how to underline, so I wrote to Tasman for help and within seven days I had the answer and all was well. I have not had any problems with Interface 1, Microdrive, Tasword 2, the Transform Keyboard or the Printer.

My reply to the statement 'Don't you think a new typewriter would be better?' is that I do not have to use Tippex anymore and amendment of mistakes is easy with Tasword 2. I can also store a balance sheet on Microdrive and by making a few simple amendments, I can bring it up to date for each bi-monthly meeting of the organisation of which I am the Treasurer. Prior to this, I have had to type out the complete

Balance Sheet for each meeting.

Finally, I can recommend Transform for quick service and a very sensible approach to customer relations. I also bought from them a Microdrive cartridge box and a dust cover for the keyboard; this was not immediately available, but was sent to me about 14 days later, after they had checked that I still wanted it.

Yours faithfully,
Major B.K. Whitworth
Salisbury

Decimal places

Dear Sir,
I have just been reading this month's edition (Aug/Sept) of ZX Computing and was pleasantly surprised to see my letter regarding decimal places had been printed.

However, on checking the listing of the demonstration program I noticed an omission in line 80, this should read: LET X = INT ((10 P) * N)/(10 P).

I also noted that my name had been spelt with an 'n' instead of a 'p'.

Regarding the problem of storing several screen images (Pen Pals Aug/Sept issue) J. Masters could probably make use of an excellent article on this subject by Duncan Overton in the Feb/March 1984 edition of ZX Computing.

Yours faithfully,
J.M. Thorp
Swindon

Chatterbox

Dear Mr Elder,
Please find enclosed a copy of my letter to Julian Chappell — you will see that I have asked him to include a note about my program in the next 'Chatterbox'.

I thought I had better let you know as well in case you get any comeback from disappointed readers.

Having just received the latest copy of ZX Computing, I was flattered that you had considered my program 'ZXylophone' worthy of inclusion in your ZX Chatterbox column.

Unfortunately, in the transition from the SFSC Newsletter to the hallowed pages of ZXC, an essential line of text went missing, namely — 'Set your machine in FAST — the program will not work in SLOW.'

I am concerned that readers may not realise this and will be disappointed that their 81s remain mute. Please ensure that a 'correction' is included in your next column.

I have been getting a favourable reaction to the sound programs I have written — did you know that the sound can be channeled into a hi-fi through the MIC port? I am at present working on a 'sequencer' which stores played noted for replay. We shall no longer be a 'silent minority'!

Congratulations on your column — it should reach the parts other publications don't reach!

Keep up the good work!

Sincerely
Paddy Moindrot
Shropshire

Microdrive files

Dear Sir,
I am writing to you because I think the following information may help with the problem of identifying what type of file has been saved to a ZX microdrive cartridge.

The problem arises when you CAT the microdrive; the CAT does not tell you what type of file(s) they are, whether:

a. Program file
b. DATA file
c. SCREEN$ file
d. a CODE file

While I was fiddling with SAVEing a file in different colours I noticed that I could SAVE programs and file with all sorts of colours. But of course the file name should not exceed 10 bytes. If while SAVEing a file you restrict the actual name to eight characters and the next two bytes to either one of the following: DATA, CODE or SCREEN$.

So, when you are SAVEing a file you would do the following:

SAVE * "M";1;"namexxxxx DATA" DATA ( ) (where DATA in the quote marks is extended mode D.)

"M";1;"namexxxxxSCREEN$ "SCREEN$ (where SCREEN$ in the quote marks is extended mode / symbol shift K.)

SAVE * "M";1;"namexxxxx CODE"CODE Address, length where CODE in the quote marks is extended mode I.

So, when you do a CAT the following information is displayed:

CARTRIDGE NAME

namexxxxx if a program
namexxxxDATA if a Data file
namexxxxSCREEN$ if a Screen
namexxxxCODE if machine code

Of course when LOADing a file from cartridge you have to go through the same procedure except that you use LOAD instead of SAVE.

I hope readers find this information useful.

Yours sincerely
Omer Coruk
London

Kempston E

Dear Ray,
I have some comments to make on your April/May edition.

Where was the latest edition of light screen designer? I would really like to see this finished!

Mr Tydeman also asked how to send multiple printer codes with the Kempston E interface. This is easy, enter either as a line of program or a direct command COPY:REM CHR$ 0 this stops the interface sending its own control codes, leaving the user in control. All that is now needed is for the user to enter as part of his program the lines of basic detailed in his printer manual for the facilities required.

For example, for my Epson RX 80 entering the line

LPRINT CHR$ (270):CHR$(14);

will switch on the enlarged print mode.

In your article 'A Present from Uncle' you gave a method for altering the address at which the m/code works from by altering the absolute jump address. Why not change the instructions to relative jumps. Then the routine will work from any address without further changes.

The Z80 JR instruction allows a program to jump up to either 127 bytes forward or 128 bytes backwards. This is quite adequate for the large print routine. To make the alterations LOAD the routine to its normal address of 32256 and then enter the following POKEs:

POKE 32341,24 POKE 32342,172 POKE 32343,0	The JR instruction jump minus 84 bytes now spare!
POKE 32361,24 POKE 32362,181 POKE 32363,0	jump minus 75 bytes
POKE 32409,24 POKE 32410,149 POKE 32411,0	jump minus 107 bytes

Why the routine was not written like this to begin with only Psion know. The routine can now be saved and loaded back to any address. Just call the routine from the loaded address and hey-presto it works!

The program 3 that you provided to alter the m/code does not work! In line 10 you INPUT s and then CLEAR s-1. This makes my issue 2 Spectrum forget the variables! Giving an error code. Was this routine tested?

Thanks for the good mag and please give us more light screen designer!

Yours sincerely
Ed Corteen
Wiltshire

What does it do?

Dear ZXC,
I want to say a few things about your 'What Does It Do?' article in the Aug/Sept issue. I am almost 12 and I have learned a lot. I was always baffled when I turned on the TV and saw people talking about 'sin. . blah. . cos. . waffle'. But I now understand these things. I like trying to program in m/c and BASIC. There's helpful software eg Collins Gem, and you can make some very good software for yourself.

So thanks Sinclair for your popular inexpensive, brilliant computers. If you're sick of Auntie Ethel's yak, then open up a briefcase and beat the hi-score on Jetpac. Anyway, hard keys hurt my fingers and remind me of a certain computer beginning with a 'B'. I like soft keys, especially for playing games with.

Brian Gillespie
Crownhill
Plymouth

Mancomp

Dear ZXC,
I took my Spectrum 48K and Interface 1/microdrive to my local computer repair shop, but they were unable to help me in any way.

Eventually I decided on Mancomp in Manchester and, after a ▶

short visit my Spectrum was working faultlessly.

I would like to recommend Mancomp to all Spectrum owners.

Yours faithfully
Andrew Duncan
Glossop, Derbyshire
p.s. They make a good cup of tea as well.

Cartoons

Dear Sir,
I am a ZX Computing reader and enjoy your magazine very much. I particularly enjoy your amusing, topical cartoons. I enclose my own cartoon which I think you may like. I would be delighted if you would publish it.
Yours sincerely
Una O'Connor
Co. Kildare, Eire

And we're delighted to publish it. Una's cartoon is printed elsewhere in the letters page.

Avast behind

Dear ZXC,
Well shiver me timbers! Your typesetter must have been as anxious as you to get away on holiday. I fear you are in for some confused readers with regard to figure 1 ('Program Layout') in my program 'Voyage of Peril' (ZXC Aug/Sept).

The following corrections are required to reflect the original:

1. Add an arrow from box 100 to box 180.
2. Box 230 should read 'Check course with wind'.
3. Add an arrow from box 230 to box 1200.
4. Delete arrow head at bottom left of box 330.
5. Add an arrow from box 310 to box 1400.
6. Box 440 should read 4400.
7. Add an arrow from box 3800 to box 4400.
8. Add an arrow from box 3900 to box 4400.
9. Add an arrow from the 'Trade Routine' box to box 4400.
10. Delete the horizontal arrow to box 1645.
11. Add a horizontal arrow to box 1600 from the far left.

The rest of the article is splendid so belay keel-hauling. . . the cat o' nine tails should be sufficient!

Thanks for a super mag.
Norman Brooks
Snodland, Kent

Oh well. . . actually to be fair, it had nothing to do with our typesetters (who are a splendid bunch of people, especially since they can cause us all sorts of problems if we don't keep them happy). In actual fact it was all Jerry's fault — Jerry is the chap who does all our artwork for us, and we don't have to keep him happy at all. So, memo to Jerry: IT'S ALL YOUR FAULT!!

Hex v Dec

Dear ZXC,
After reading the letters in the Aug/Sept issue of ZXC, I feel that I must reply to two of the letters that were sent in to you.

First, the reason why listings are sent in hex is that it's shorter. For example, AF B0 3C — three bytes in hex takes six characters, whereas 175,176,60 — three bytes in decimal takes eight characters.

Not only is it shorter to write machine code in hex, but hex always stays the same length, so you can write long statements like A6 B4 C0. In decimal, because the length varies (ie 201 or 15 — one has two digits, the other three), you must divide them with commas ie, 166,180,192. So, hex really is more convenient.

As for D.J. Shellhorn's letter about sound, six and eight channel sound can be made by linking the three channel sound and I/O interfaces from William Stuart Systems. Just one of these intefaces will provide three channel sound and sixteen I/O ports. Two units will provide six channel sound and thirty two I/O ports, while three units will provide nine channel sound and 48 I/O ports. However, no more than three units can be linked together.

I finish by congratulating you on your excellent magazine.
Yours sincerely
Simon Bryan
Nr. Newbury, Berks

'81 tips

Dear ZXC,
In reply to G.M. Raines, I would be happy to swap ZX81 adventure tips if he would like to contact me.

Meanwhile, here are a few tips for the Artic range of adventures:

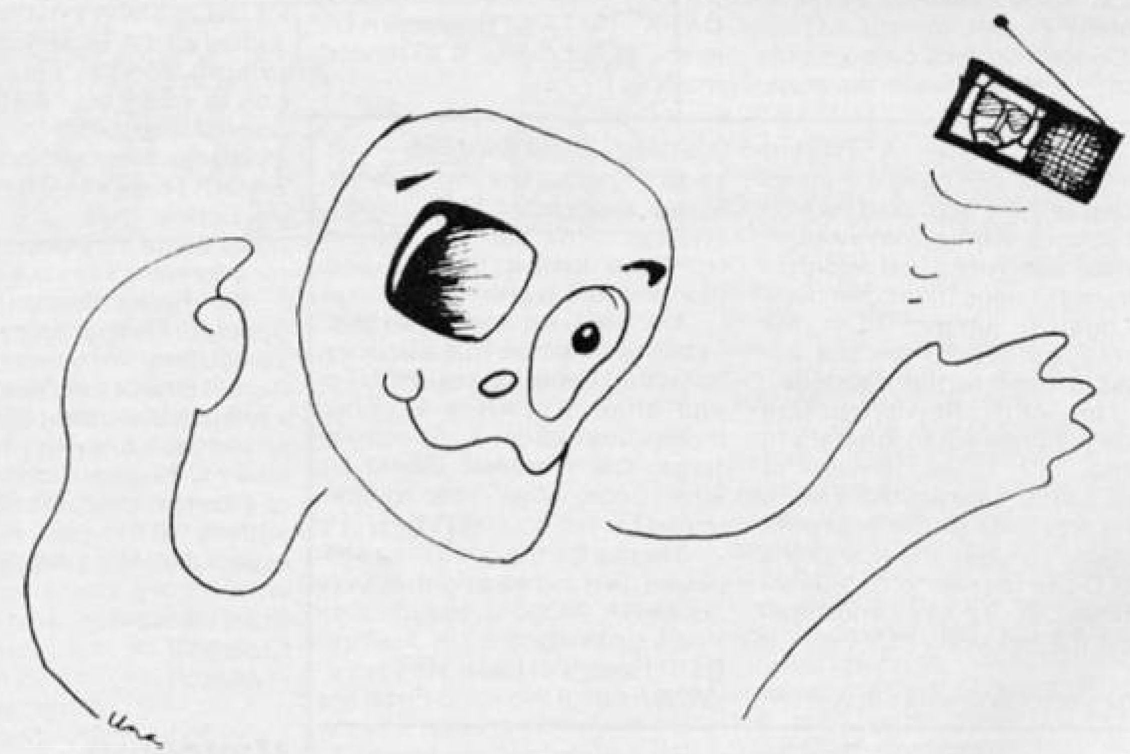
Planet of Death — wear the boots to retrieve the gold coin from the lake. Guards can be bribed. After firing at the force field, dance to the music playing on the radio.

Inca Curse — in the fire room, get the lamp before putting out the flames. Ladders are very helpful to get onto high ledges.

Ship of Doom — silver rods + battery = sonic screwdriver. Improve your eyesight with glasses. Don't trust little girls. Fighting is better than running away. Look up frequently. Rope + hook = help.

Espionage Island — the first problem is solved by feeling the dark corner of the plane wreckage and then pulling. Unstrap parachute, do not go south, go west twice. Give the beads to the native woman who will drop a knife at your feet. The graffiti is important, so read it. Stitched is the same as sewn.

I would also like to hear from ZX owners who use their machine for programs other than games.
Yours sincerely
Mark Eldon
146 Fairfield Road
Heysham, Lancashire
p.s. I would like to know if any of your readers have any of the following issues of ZXC that they would like to sell: Volume 1, nos 1,2,9,10,11: Volume 2, nos 1,2.



AND NOW FOR THE LATEST IN SINCLAIR MINIATURE TV'S. . . . . THIS LITTLE BEAUTY IS WORN AS A CONTACT LENS!

VISA ACCESS

SPEEDYSOFT

01-846 9353 (24 HRS)

EUROCARD MASTERCARD

WE ALWAYS TRY TO SEND YOUR PROGRAMS ON THE SAME DAY WE GET YOUR ORDER, THAT'S WHY IT'S CALLED SPEEDYSOFT! All programs normally in stock before we advertise them! FOREIGN ORDERS WELCOME, satisfied customers in 64 countries for 3 years. UK prices include VAT: export prices are the same, plus p&p, because of the extra work involved. Pay by sterling cheque, bank draft or postal order. All orders must be prepaid: we send your programs as soon as we get your money! It's faster to order by phone from any country with your VISA, EUROCARD, MASTERCARD or ACCESS. Call 01-846 9353, 24hrs, 7 days a week. ALL PROGRAMS FOR SPECTRUM AND SPECTRUM +. BIG FREE ILLUSTRATED CATALOGUE WITH EVERY ORDER: DETAILED REVIEWS, LOAD TIMES, WHAT THE CRITICS SAID, ETC.. Catalogue includes all the classic 'serious software', too. Send £1.00 (refunded with your first order) for catalogue only. TWO FREE BLANK TAPES OR ONE FREE MICRODRIVE CARTRIDGE IF YOUR ORDER FROM THIS AD COMES TO £30 OR MORE, excluding p&p. State choice on order form.

". . . A VERY PROFESSIONAL SERVICE AND LIKE A BREATH OF FRESH AIR"
(G. Mills, Surrey)

HITSQUAD!

'THEY SOLD A MILLION' HAS TO BE ONE OF THE BARGAINS OF THE YEAR! On one tape you get four great programs for the price of one. BEACH HEAD from U.S. Gold. JET SET WILLY (the original version) from Software Projects. DALEY THOMPSON'S DECATHLON from Ocean. And SABRE WULF from Ultimate. Original prices totalled over £30. (Hitsquad) Cassette £9.95

AMX MOUSE

At last the famous mouse runs on your Spectrum. The package includes your mouse plus four programs. **AMX ART** a drawing program full of icons, pull-down menus, pointers. Not just for fun, but OK for engineering/architectural drawing, too. **AMX COLOUR PALETTE** lets you fill in with colour any drawing from AMX ART. **AMX CONTROL** allows you to use the mouse in your own programs. **ICON DESIGNER** creates and stores icons for use in your own programs. The whole package, with manuals, just **£79.95**

THE UNICORN

*"Offers more than just larger programs ... it's all most sophisticated and when you consider that there's also a built-in joystick port, it really does look like Mikrogen have made a breakthrough." (Crash)*The first ever Spectrum program to come with its own interface, to turn it into a 64K game! And there's a head alignment routine, too. Load/save to/from tape/microdrive. 104-page illustrated paperback with clues for the adventure. Map of the game. Even a Helpline phone number. And 6,500 locations to explore in this graphic adventure. (Mikro-Gen) **Cassette/Interface £14.95**

WORM IN PARADISE

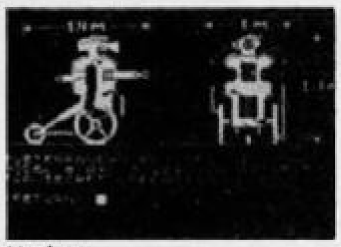
A brand new adventure system from Level 9 is a major event! More text than ever, by compressing ordinary English over 50%. Vocabulary over 1000 words! Understands more commands than any other adventure. Plus 200 pictures and now, with multi-tasking, the game does not stop while the picture is drawn. No more waiting! Amazing. (Level 9) **Cassette £9.95**



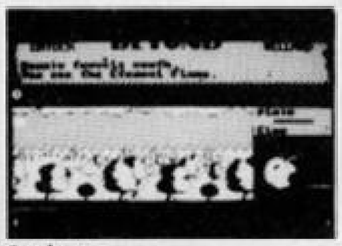
Nato Alert



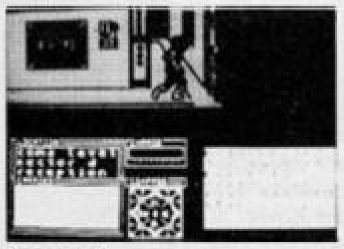
Shadow/Unicorn



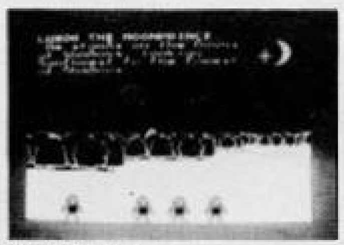
Hacker



Sorderon



Marsport



Now Games

NATO ALERT

A wargame simulating a non-nuclear conflict in Europe between NATO and Warsaw Pact forces. 1 or 2 players, 5 skill levels. Up to 21 different objectives may be chosen, to determine length of play. Complex, horrifyingly realistic. Map graphics help. Hints given, too. No Stick. (CCS) **Cassette £5.95**

HACKER

"Hacker is not like other computer games. We've told you how to load the program. Everything else is up to you. A real-life hacker would proceed by trial-and-error and that's how you play this game." That's what the inlay card for 'Hacker' says. So we won't tell you how to find out about Magma Ltd, or where your SRU's phlasmon valve is, or even what your vital SRU is. But there are good maps, lots of animated graphics and plenty of messages! The more you discover, the more you enjoy it. (Activision) **Cassette £8.00** While you're at it, why not buy yourself the 'HACKER'S HANDBOOK (150-page paperback) for just £4.95 as well?

MARSPORT

*"This is what the kids have been waiting for, an adventure game with destruction as well! The best yet from Gargoyle!" (Popular Computing Weekly) "Just watch John Marsh shoot ... it is one of the best animation sequences ever seen on a Spectrum!" (Crash)*Dun Darach in space? Not really, but Commander John Marsh has the same walk as Cuchulainn. Brilliant animation, with much improved 3D. DEMO and SAVE available. (Gargoyle) **Cassette £9.95**

NOW GAMES

*"Excellent value ... should keep you busy for months ... if you're new to computing, you're laughing." (Popular Computing Weekly)*This compilation tape contains: **LORDS OF MIDNIGHT (Beyond): BRIAN BLOODAXE (The Edge): PYJAMARAMA (Mikrogen): STRANGELOOP (Virgin): FALCON PATROL 2 (Virgin): ARABIAN NIGHTS (Interceptor)** It is astonishing value for money. (Virgin) **Cassette £8.95**

SORDERON

"Combines the best of text adventuring with the highly successful landscaping techniques to produce a game that sets new adventure standards." (Computer Gamer) "A true successor to the realms of Midnight ... definitely shows the way forward." (Sinclair User) 4,100 locations. 750 word vocabulary. 3 sentence command recall. Now you can see every step of your quest and talk to the characters you meet! Cheat Sheet available from Beyond if you get stuck. Score. And SAVE. (Beyond) **Cassette £9.95**

"YOU'RE GETTING BETTER WITH EVERY CATALOGUE"
(K. Volkmar, W. Germany)

MONOPOLY

"An excellent simulation ... vastly superior to the other, non-authorised versions ... the graphics are very realistic." (HCW) New Official Monopoly, with 3D board view; for 2 to 6 players any or all of whom may be the computer. Specify length of game if you wish. No Stick. (Leisure Genius) **Cassette £9.95**

BORED OF THE RINGS

"An excellent and vast parody ... Challenging and atmospherically written." (Popular Computing Weekly) Enhanced version of this huge, funny, beautifully graphical adventure spoof loads in 4 parts. No Stick. (Silversoft) **CASSETTE £6.95**

POST TO: SPEEDYSOFT (ZX15)
37 CHURCH ROAD, LONDON SW13 9HQ, ENGLAND.
For CATALOGUE ONLY, send £1 cash. Refunded with your first order.
I own a SPECTRUM. I enclose a cheque/PO payable to SpeedySoft
OR charge my VISA/ACCESS/EUROCARD/MASTERCARD

No. ☐☐☐☐ ☐☐☐☐ ☐☐☐☐ ☐☐☐☐

Signature: .... Expiry Date ....
Please write clearly. If we can't read it, you won't get it.
Name: ....
Address: ....
....
.... Postcode: ....
PHONE NO: if any, in case of query ....

Program Name	Price
Postage & Packing UK add 75p per order	
Europe ADD £1.00 per program	
Outside Europe ADD £1.50 per program	
Total Order	

Shoptalk Shoptalk

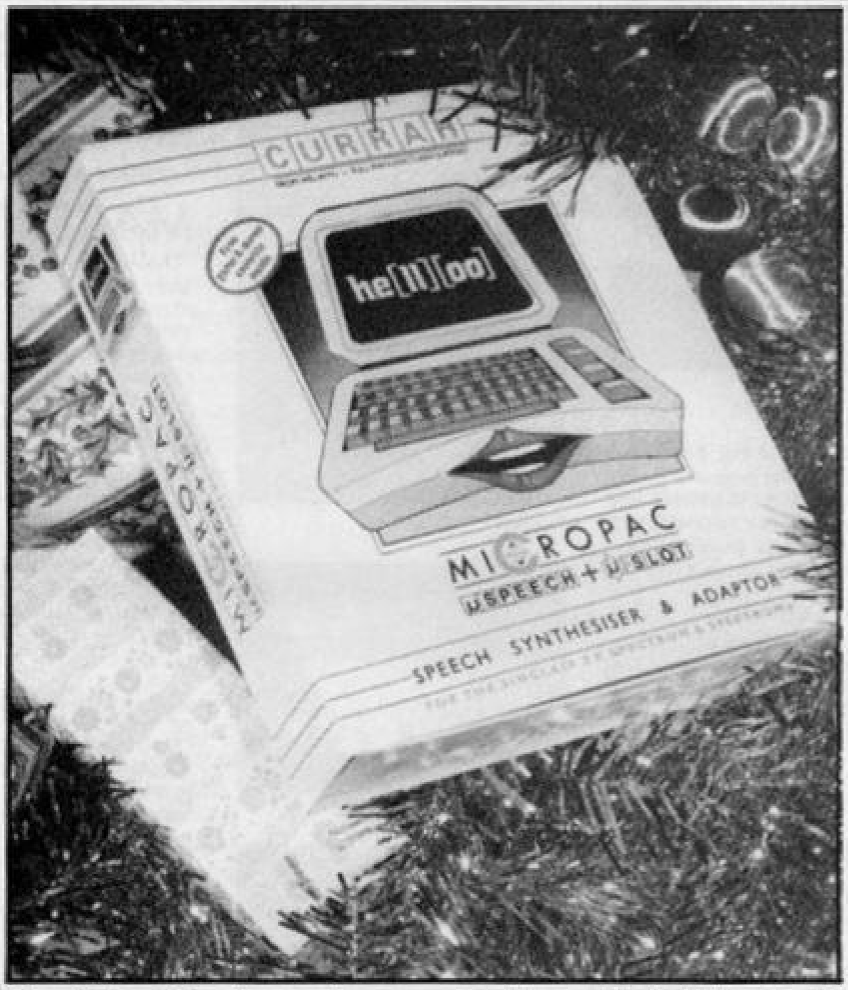
Odds and ends, letters, and company info

A letter from Eclipse Software

Dear Ray and Cliff,
Thanks for the review of Letaset in the August issue, but your review in the same column for Scisoft's Astronomy disappoints! The program does not mention Halley's Comet and is in fact a rip off from my book *ZX Spectrum Astronomy* which Scisoft now admit and claim (in the trade press) to have withdrawn.
Yours sincerely
Maurice Gavin RIBA FRAS.

Fighting words indeed, and perhaps Scisoft would like to reply. Also I would like to point out that in the News section in which we printed the items we are not reviewing products but simply passing on information supplied by the companies. If a review sample is sent then a comment may well be made and a full review, either elsewhere or in a subsequent issue, may be written if we can find space to include it.

Finally I mentioned Halley's Comet as a reason why people may develop an interest in Astronomy and unintentionally introduced the implication that the program dealt with it. Sorry, sackcloth and ashes time again!



The Welwyn-Currah speech synthesiser is an old favourite which is being repackaged for Xmas. If you fancy experimenting in this area it is a good buy and the fact that sound is routed through the TV will add to the effect.

Race round Britain on Micronet 800

Not an actual county connection, but a recently introduced game on the Micronet 800 service.

Developed by Mike and Lindsay Brown (Mike is the mind behind Chatline and Gallery), it is a simultaneous multi-player game in which players travel around the UK searching for four secret locations. Clues are flashed onto the screen and as the race progresses your geographical position is indicated by a flashing cursor.

Regular £100 cash prizes are on offer to give you a chance to recoup some of your telephone bill. I haven't been on-line for some time now but I will check this out as soon as I can find my modem.



BUD for RAM

RAM Electronics have taken over the BUD Computers range of peripherals. Their news release also informs us that they are about to release a new interface for the Spectrum incorporating joystick port, reset button, power indicator and expansion slot. This should be on the market in time for Christmas but no price has been quoted.

QL books from Longman

The latest in the Longman 'Working with the Sinclair QL' series is WORD PROCESSING WITH THE QL written by David Dempster. The book is aimed at three possible users of the Quill program supplied with the machine, absolute beginners, experienced computer users and experienced typists.

To enable the author to cover all these areas the book is written in a very simple manner and supported by many illustrations. If you have problems in getting to grips with the instructions in the manual then this book will supply the answers. I have been using WP's for some time now and I was able to learn a few things from the book. Chapters include Introduction, Getting Started, Basic Editing, Filing on Microdrive, Formatting, Advanced Editing and Printing Techniques. There is also a comprehensive appendix section.

Mr Chips at home

Peter Ellis went through the usual gamut of confusion and frustration when learning to program and decided to produce a series of instructional programs to help others avoid the problems he had. HOME TUTOR is the result and consists of thirteen tapes dealing with such topics as basic instructions, program design, records, files, logic, peripherals etc.

If you are an absolute beginner and wish to learn in this manner then contact Fearless Software Ltd. We were not supplied with an address but ringing

01 402 3401 should find someone who will be able to help you.

The course costs £9.80 per month plus £1.15 p&p, alternatively it can be supplied quarterly for £30.00 a quarter.

Replicating reality

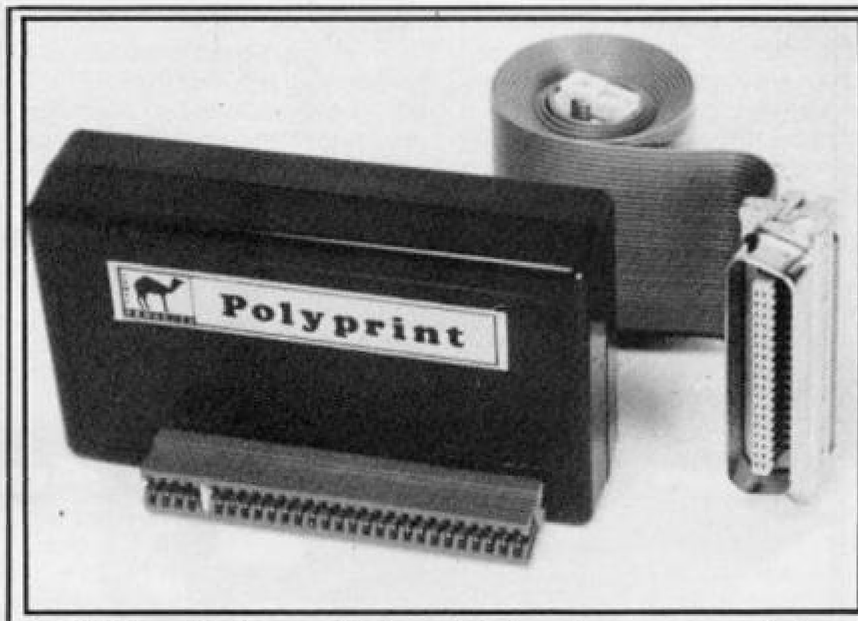
An issue without a release from Interface Publications and Tim Hartnell is rare and this time presents us with Replicating Reality, Exploring Computer Simulations. This is a discussion of the simulating of real life situations and a set of programs which demonstrate various types of simulations. This will be a useful book if your interest lies in this field, especially if you are a relative newcomer to the computing scene.

There appears to be nothing startingly innovative in the book and the concepts of most of the programs have nearly all been aired before, however it is useful to have a collection of all the same type of programs and there is plenty of advice on how to construct your own, providing, of course, you are familiar with programming in Basic.

Chapters and programs include Power of Simulations, Components of Simulation, Feedback in Action, Simultaneous Equations, Life, Robot Logo, Stock Market and many more. At £7.95 it is worth considering if you're interested in this field.



Another Xmas goodie is the Joystick and interface combination pack offered by Vulcan Electronics for around £10.00! We've used the Gunshot joystick which is part of the package for reviewing games for about a year and it's stood up to all the punishment we could dish out.



Getting into print

Camel Products have been around a while and have gained a reputation for producing specialist products. The latest item from them is POLYPRINT, a Centronics printer interface for the Spectrum. The difference between this and other Centronics interfaces is that the operating system is on Eprom which saves loading a special program.

Although the Kempston E and ZX Lprint III also have an inbuilt operating system, Polyprint's extra special feature is that it will allow text to be displayed on screen and printed in any of seven international character sets, French, German, Danish, Swedish, Spanish, Italian and, of course, English. These character sets match the Epson FX80's preprogrammed ones. Polyprint provides Llist, Lprint and Copy and is fully compatible with Tasword 2.

They do mention that if the preprogrammed character sets are not enough then you also have the facility to modify and design your own. Polyprint is £51.69 complete with cable and inc. VAT and p&p from Cambridge Microelectronics Ltd., One Milton Road, Cambridge CB4 1UY.

More joy with Mach 1

Will Cheetah ever live down their statement — 'the joystick is dead'? (Will we let them?)

Anyway, from Cheetah comes the Mach 1 joystick which looks rather nice, and it is claimed that it will withstand immense punishment. I would have loved to put this to the test when reviewing Supertest, and maybe we'll get time to try one out for the next issue.

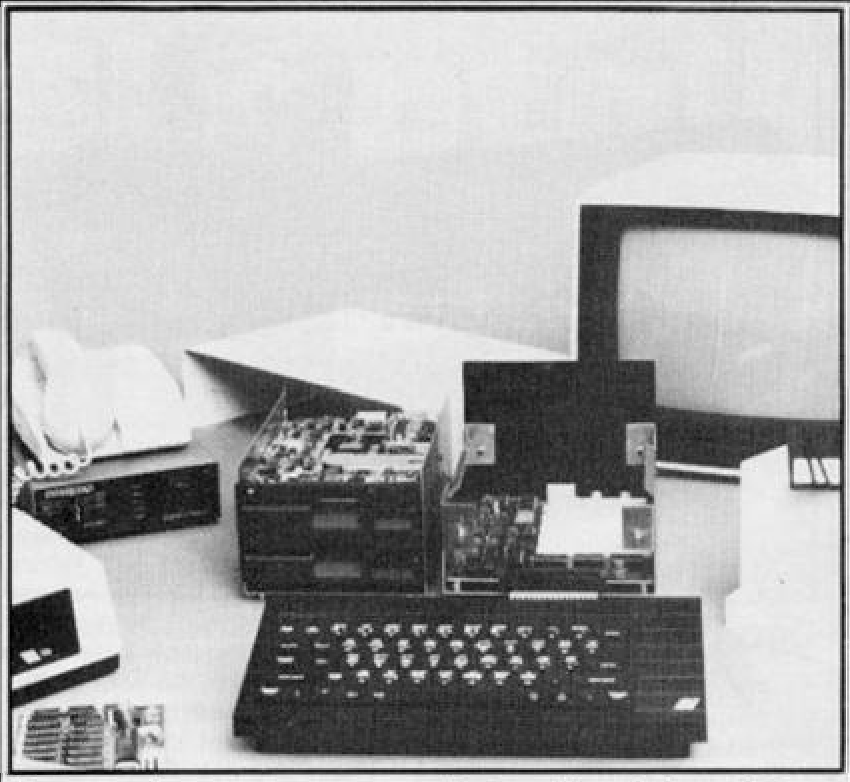
In this time of an almost confusing choice of joysticks the company claim that the three fire buttons make it worth chosing in preference to any other. A built in auto-fire switch is also featured. Look out for it at your local store priced at a reasonable £8.95.



The Saga continues

I was full of praise when I reviewed the Saga 1 Emperor keyboard from Saga Systems Ltd., and now that they have launched their superb looking Saga 3 they are reducing the price of the original to £39.95.

The look and feel of the Saga 1 made it one of my favourites to use here at the office and many an issue of ZXC was typed on it. The other two keyboards in their range are the Saga 2 Profile at £49.95 and the top of the lot Saga 3 Elite at £79.95. ▶



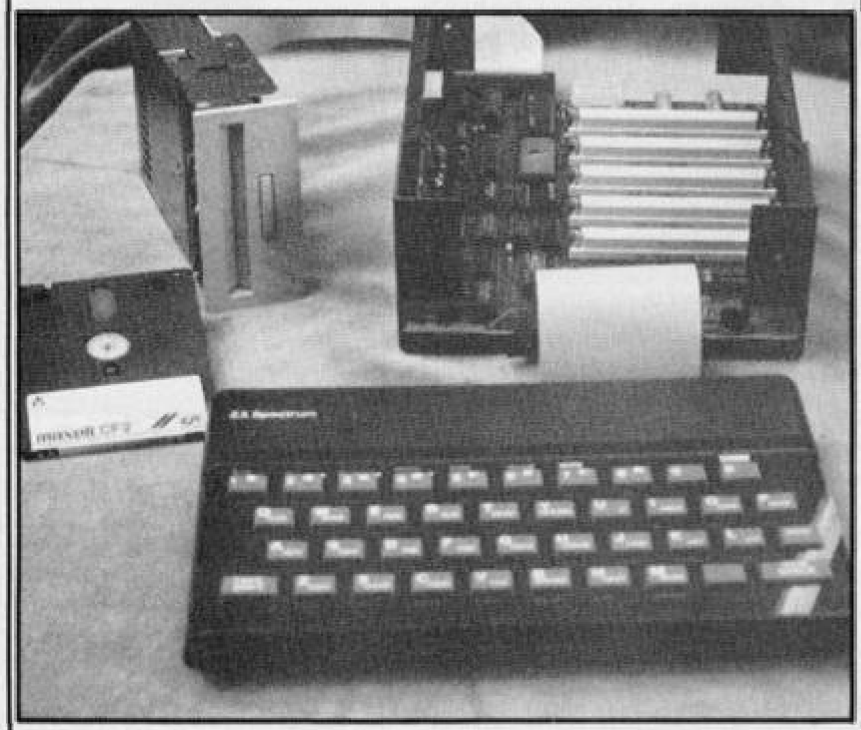
Framed

The Gordon Microframe is one of the few peripherals designed to be expanded and developed in the style of the early pioneers of computing. In itself it is an open frame disk drive controller with connectors for a range of five special cards to be plugged in at will, these give a wide and versatile application to your computer.

OK, so it doesn't look as tidy and neat as most of the peripherals on the market, but if you are one of the many hobbyist computer users and want to get the most out of your machine then this system is worth looking at. The cards available to plug in at the moment include a Centronics interface, RS232 Modem drive, D/A and A/D converters, 64K paged map memory and there are more under development.

The main purpose of their contacting us this time was to let us know that they have now implemented Interface I and Microdrive syntax, including Hook codes, to their Floppy Disc Interface. They tell us that all programs which will work with the microdrives will now work with their system.

The basic Microframe disk controller and chassis will cost you £149.50 and cards are priced between £10.00 and £64.00. If you are interested phone them on 0292 280467, or write to Gordon Micro Ltd., 3 Callendar Road, Heathfield Ind. Centre, Ayr KA8 9DJ.



Seeing stars?

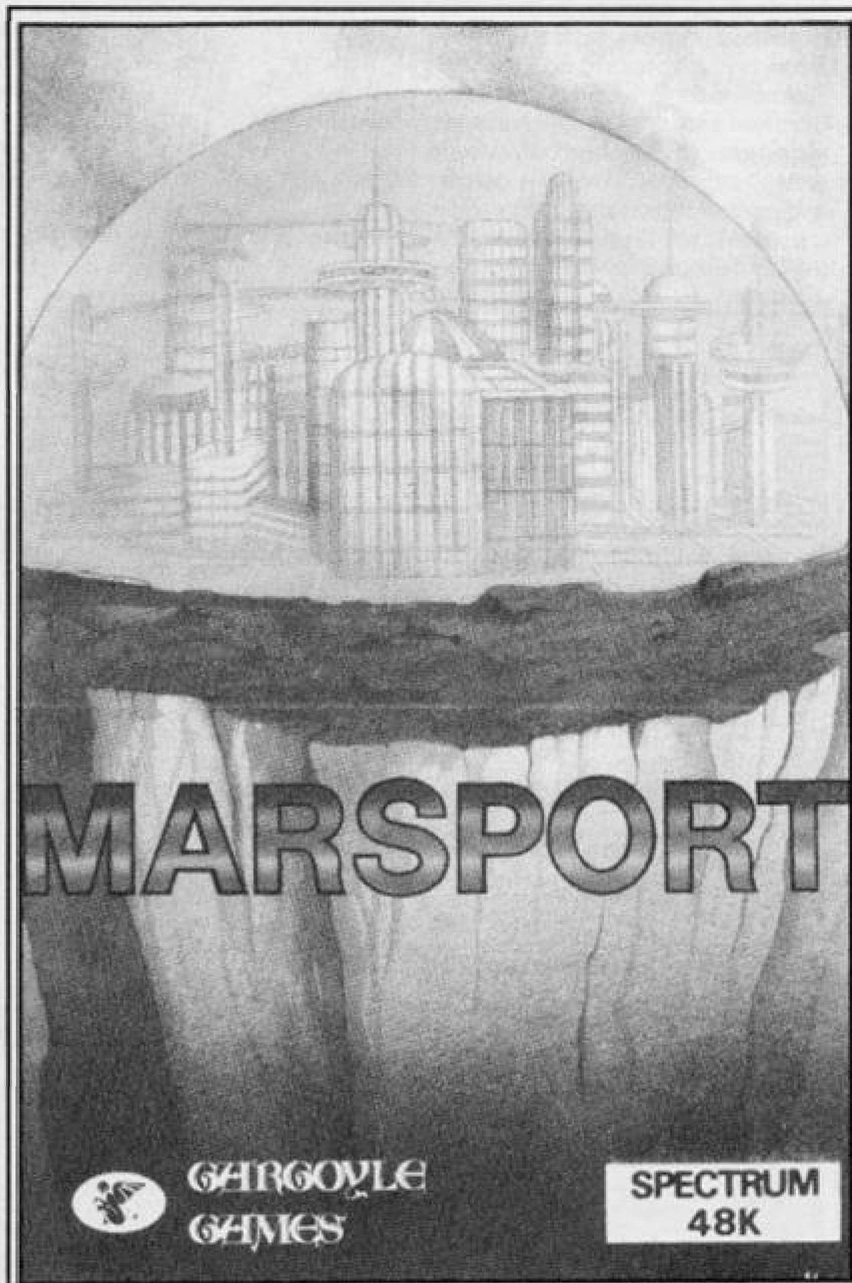
Eclipse Software have kept us informed of their products with gratifying regularity, and they have added three new programs to their range.

HALLEY'S COMET is the only program we know of dedicated solely to the comet and appears to be the definitive computer work on the subject. £6.95.

ASTROLABE. Armed with a telescope and a Spectrum and this program you have a vast and accurate amount of information about the sky you see on any date. The perfect way to combine two hobbies. £6.95.

CHEMICAL FORMULAE is a databank and revision aid all in one. O and A level students should find it useful. £9.95.

Contact them at 79 Ardrossan Gdns, Worcester Pk., Surrey KT4 7AX.



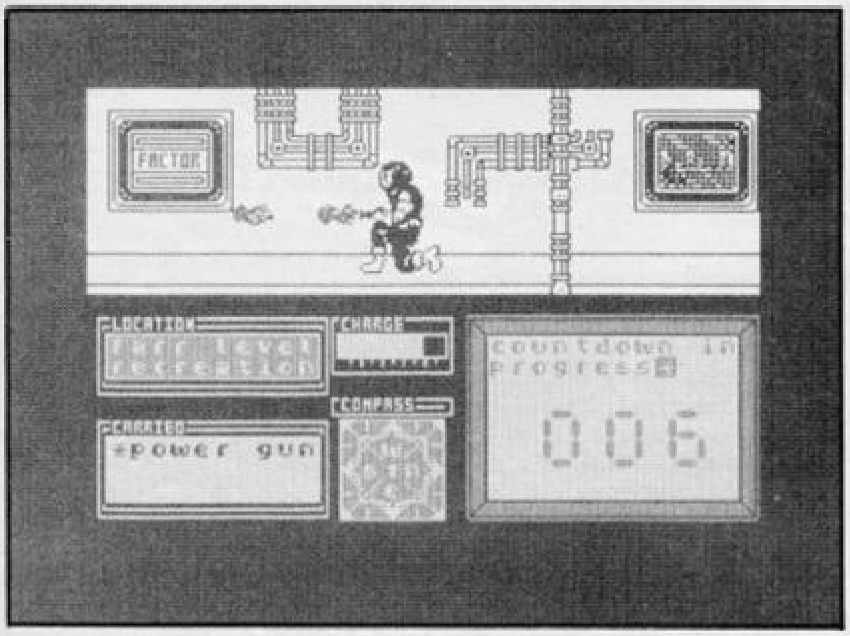
Under seige

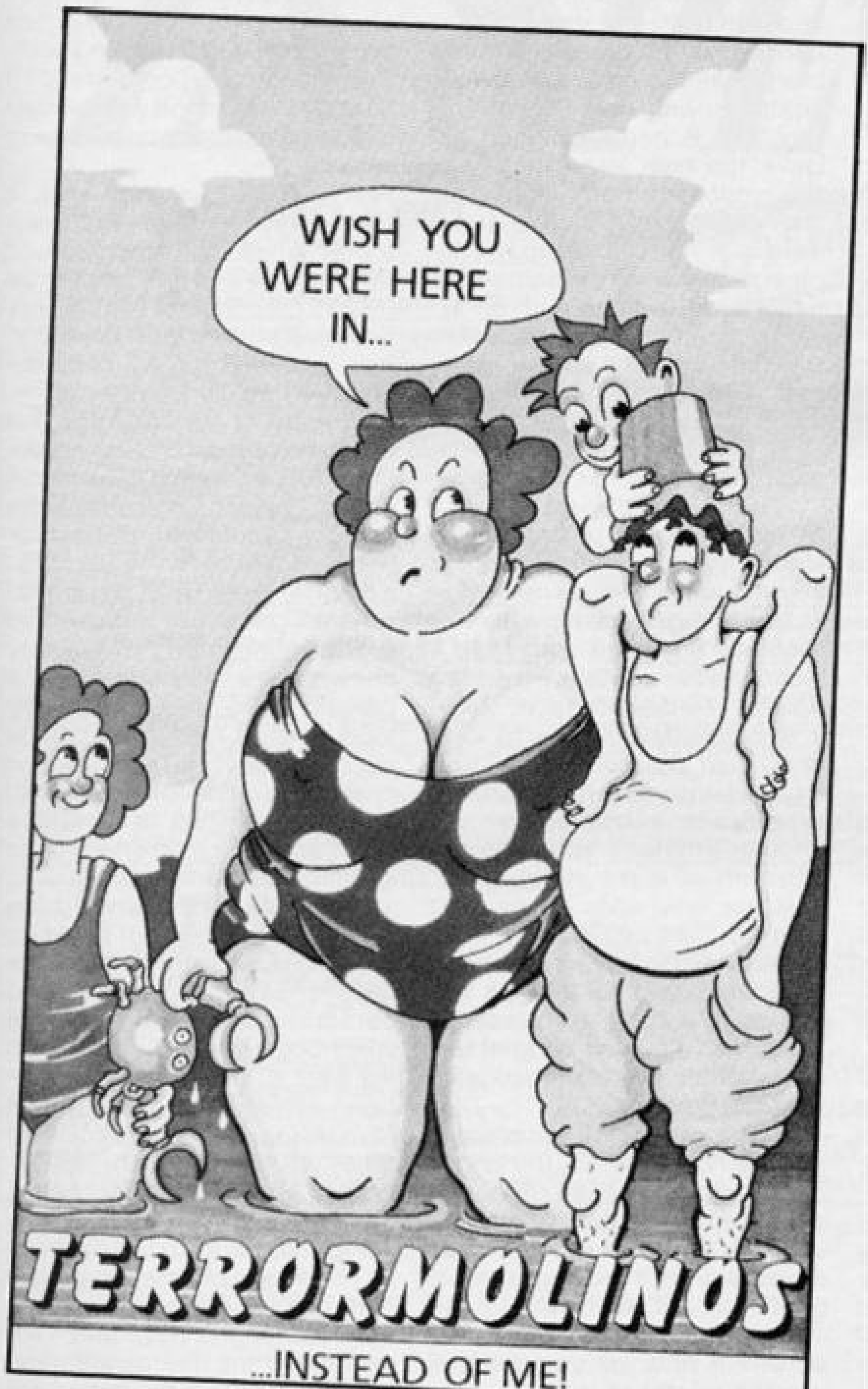
Forget the Alamo, the whole Earth is under seige in Gargoyle Games latest three part series of programs. Covering 500 years from 2424, the first part, MARSPORT, has you taking part of John Marsh, searching the enemy occupied city of Marsport to find the plans which will strengthen the forcefield which is all that prevents the end of civilisation as we know it.

Described as a graphic adventure featuring the 'most sophisticated animation ever produced on a home computer' it has ten levels, 920 corridors and a host of puzzles and aliens.

Parts two and three, Fornax and Gath, are due out in January and April '86, and each part costs £9.95. We have all heard similar claims before but at that price they're going to have to deliver the goods.

If you can't afford Marsport then perhaps SWEEVO'S WORLD may fill the gap, Described as a funny arcade adventure, it features Sweevo the clumsy robot. Sounds interesting, at £7.95.





Melbourne mania

Melbourne House has earned a lot of respect from us punters, but new programs from there are an infrequent occurence, so the programs that they have released since our last issue are a welcome surprise.

TERRORMOLINOS another off beat adventure from the authors of Hampstead and I am sure it will be a hit — probably by the time we go to press — it features the same sort of humour as Hampstead and although I only loaded it to give it a quick run through I was caught up in its atmosphere and ended up playing for over two hours.

The post card style graphics are an interesting attempt at being different but I felt that the game, though improved, would have still been excellent without them. £7.95.

FIGHTING WARRIOR is similar graphically to Exploding Fist, but your character uses a sword and has a complex quest to undertake. We review it fully in The Jury.

WHAM — THE JUKE BOX, has something to do with the group of the same name, and even though we saw an early version there is enough here to state quite simply that it is the BEST program for the musically inclined Spectrum user.

Providing two simultaneous sound channels from the software alone, plus rhythmic white noise options, graphic representation of musical notation, tempo, editing, looping and the ability to save tunes in a compiled form for use in your own programs, either on its own or at the same time as a game is played. Unfortunately we haven't the time for a detailed review as it really needs a week or so to do it justice. I will give it a darn good going over and report in the music section next issue. £14.95.

Software news .... In Brief

• For schools and colleges everywhere the information that a group of major educational software houses have joined together could well be a cause for celebration.

BESA — British Educational Software Associates — has been formed to improve the distribution and supply of educational software. Some of the companies involved are ASK, Collins, Widget and Griffin. For more details write to BESA, 4 Little Essex Street, London WC2R 3LF.

• Another group have joined together. Ocean, Software Projects, Ultimate and US Gold have created a compilation tape of their oldest, oops, greatest computer programs. Called HITSVILLE and priced at £9.95 this does actually represent good value for money, especially if you are a relative newcomer to the computing scene. The programs are Beach Head, Decathlon, Jet Set Willy and Atic Atac.

• Adventure International's latest offering brings fiction's most durable hero to the computer screen. ROBIN OF SHERWOOD — THE TOUCHSTONES OF RHIANNON is based on the recent TV series and follows the established format of an adventure with graphics.

Sure to win as big a following as Gremlins and their other successes, this program is available at £7.95.

• I'm in two minds about Quicksilva's new game Schizofrenia. Hectic and somewhat weird arcade action from a well established company for £7.99.

• PSS produced Mcoder, then Mcoder II and now, after a lengthy pause, MCODER III. I loved Mcoder II even though you had to write in a kind of pidgeon Basic. The latest version is claimed to compile the entire list of Spectrum Basic commands and handle full floating point arithmetic.

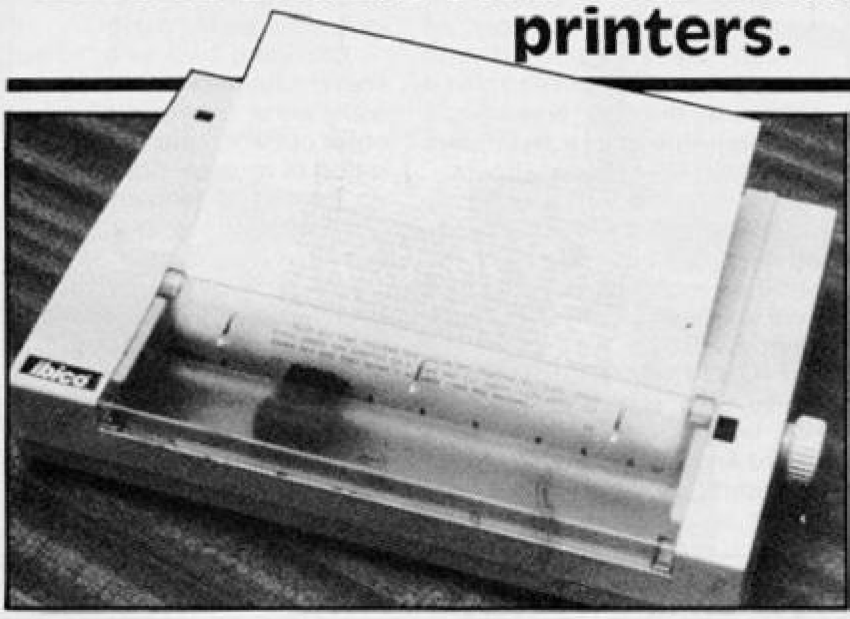
For life in the fast lane look out for it at £12.95.

• Last issue we mentioned the strange goings on at a certain St. Bride's School and their Quilled adventure The Secret of St. Brides. We enjoyed it and said that subsequent programs were likely to be exciting as they developed their style. THE SNOW QUEEN is the next and could be a biggie.

Designed to be played whilst reading the copy of the novel included, you do not control the main character but advise her. This means she will not always do as you want and may even argue and answer back! It sounds unique and we'll try and get a copy and get our intrepid reviewer to give it a good talking to.

Getting into Print

ZXC looks at two low cost, high quality printers.



Brother M-1009

Ibico LTR-1
Supplier: Saga Systems, 2 Eve Road, Working, Surrey GU21 4JT (04862) 22977

If you use your computer for anything other than beating back never-ending hordes of aliens, a printer is becoming almost an unavoidable necessity. As the QL includes the excellent word-processing package Quill in the price, owning one without the accompanying printer is rather like owning a shotgun without any shells, but its difficult even to take full advantage of the Spectrum without one.

The only problem is that printers of any quality can be horrendously expensive, often costing twice the price of the computer. Fortunately, prices have been dropping recently and — even better news — a range of low cost, high quality printers looks certain. Basically the idea seems to be to give the buyer high grade print without all the added features that, however desirable they may be, add substantially to the cost. Already two machines of this type have appeared on the market, the Brother M-1009 and the Ibico LTR-1 sold in this country by Saga Systems.

Brother M-1009

The brother machine is a lightweight (3kg), compact (330mm x 190mm x 70mm), near-letter quality, nine-pin impact dot matrix printer designed to sell at just under £200. The connector cable costs between £15 and £20 depending upon where you buy it (alternatively you can make your own). There is an optional paper roller available (which costs around £20) as well as a dust cover and a pin feed unit. The operations panel consists of two pad keys (on-line and line feed) and three indicator lamps; power, error and on-line. Printing is via a cassette-type, endless fabric ribbon with an estimated life-span of a half-million characters.

Printing is bi-directional logic seeking with a speed of 50 characters per second. The character set consists of 96 ASCII, 48 European, 16 Greek, 48 graphic and 21 mathematical and other symbols. The number of characters per line varies from 80 in the normal mode to 132 in condensed mode. There is an enlarged mode of 40cpl used mainly for headings and an enlarged condensed mode of 66cpl. Other features include superscript, subscript, italics, emphasized and double-strike modes. These various print modes may be combined to give a strikingly varied output.

The print quality is excellent. As a writer, I try to give editors polished manuscripts (but rarely succeed with a typewriter) and normally this would seem to call for a letter quality printer. But using the M-1009's combination of double-strike and condensed modes is really all any reasonable person could ask for. I suppise one wouldn't use the M-1009 to write a party invitation to royalty but it should for just about everything else including the most picky of editors (*Are you trying to tell me something? — Ed*). It's not so much that the standard of print is radically different from other quality dot matrix printers but rather that you get the same sort of quality for a lot less money.

The emphasized mode itself is perfectly adequate for virtually any day-to-day use as is the condensed mode which looks marginally more like an obvious computer print-out. The normal typeface is perfectly good but lacks the near letter quality of the emphasized and (especially the emphasized/double-strike modes. The accompanying manual is fairly comprehensive but it could well do with an accomplished Japanese-English translator to tidy up the grammar and idioms. My manual came with a set of addenda that was actually longer than the original thing.

The 'R' version of the M-1009 looks to be pretty much the standard issue in this country and is the one most likely to interest Sinclair owners as it comes equipped with both a Centronics (parallel) and an RS232 (serial) interface. While it is perfectly possible to connect most parallel printers to a QL (and with a bit more difficulty, a Spectrum), it requires yet another interface added to the labyrinth of wires at the back. This not only adds to the cost (some of the available interfaces are reasonably cheap but many are fairly pricey for anyone trying to build up a computer system on a limited budget) but it's another piece of gadgetry that can go wrong.

Some of the QL peripherals now being marketed (the Bright Star modem, for example) carry a Centronics interface as an added extra but there is still a clear advantage in choosing a serial printer to link directly into the computer. After all, if the peripheral acts up, you can still use the printer to write a complaining letter to the company. If you need to alter the DIP switch settings, they can be reached with any sort of small tool through a hole provided in the frame. Otherwise, once you get the right cable, everything is ready to go.

Ibico LTR-1

If you want high-grade letter-quality print at a very reasonable £120 (previously £169.95) then the LTR-1 is the (only) answer, but it carries few, if any, additional feature, like the M-1009, the LTR-1 weighs 3kg but is slightly smaller measuring 298mm x 198mm x 63mm. It prints at 12cps and employs a unique golfball like head mechanism that takes its ink from a roller (replacable at £2.50) situated directly behind it. The head is estimated to have a life-span of 20 million characters and carries only one typeface, Elite 12, to work with. The character set consists of 96 letters and symbols spread over a barrel-like arrangement of five wheels.

The LTR-1 comes with a Centronics interface but there is a mechanism for adapting it to directly accommodate an RS232C connector to effectively convert it into a serial printer. It is designed for A4 paper (or anything smaller) and has an annoying paper-feeding knob that has to be pushed in to turn and is hard to get a real grip on. A slightly bigger knob offering a real grip would be a distinct improvement. Nonetheless, anyone wanting straightforward computer printout and letters/manuscripts with a very high quality letter print at a bargain price should seriously look at the LTR-1. The print is indistinguishable from a good typewriter and would be ideal for anyone willing to sacrifice a bit of speed in exchange for an economical wordprocessor.

Both Brother and Ibico deserve a great deal of credit for bringing quality printing within a price range of the non-dedicated computer hobbyist and the small businessman toying with the idea of some computerisation. Of the two, Brother's M-1009 is clearly the choice for those wishing to take full advantage of the capabilities opened up by linking a printer to a good computer but Ibico have provided the main essentials at a price that more people can afford. Both companies have taken advantage of the downward trend in printer prices by opting for quality output at bargain prices. In the long term, it's the user that gets the benefits.

Brian Beckett

QL Delta Disk

Ray Elder goes to sunny Staines to investigate Technology Research's new interface.

'Hello?'

'Hello,' said a voice. 'It's been a long time.'

'Yes,' I said, not having any idea what had been a long time and wondering which of my sins had found me out.

'John Rutherford speaking, TRL.' the voice said, I breathed easy knowing we weren't going to go through the 'Bet you don't know who this is' routine.

'Why don't you come down and see our new QL disk interface?' John continued, and so, a year after I last travelled to Staines I found myself ambling west to visit Technology Research once again.

On arrival, nothing had changed outwardly, but inside the building the company had changed rooms and I once more wandered the maze that is Unit 18. When I arrived, the first thing that struck me was the neatness and comfort compared to their previous abode.

I was ushered into the main office where it was pleasant to renew the acquaintance of John, Steve Tam and Steve Mok. Also to meet C. Tang the production manager and Y. Cheuk the marketing director. Many happy moments passed discussing the state of computing in general, the QL in particular, and their interface in brief. I left with not only the Delta interface for the QL but also a new version of the Spectrum disk interface, the Beta Plus. 'There are a few surprises in that one for you,' said Steve with a glint in his eye.

I took the Delta manual home to read, because by the time I got to the office it was well past 6.00pm (overtime accountant please note) and anyway I prefer to read instruction manuals before trying things out, it can save a lot of frustration and even expense!

It took about twenty minutes to read the whole thing, I liked the simple informative manner in which it had been written — no previous knowledge is assumed — and not only are connecting and operating instructions given, but also an outline of disk handling and disk drive types. Although it helps if you have a little computing knowledge I felt that someone who had bought a QL to use for business and had no knowledge of BASIC would be able to operate this system without too many problems.

After reading the manual, even without access to a machine to try things out, I felt I could operate the system.

Theory

The Delta Interface in basic form contains the Disk Operating System (DOS), a parallel printer port (Centronics) and a RAM Disk system. Extra on-board memory of 64K or 128K can be supplied as an optional feature.

The DOS was written by one of the QDOS (the QL's operating system) authors and is therefore fully compatible with both QDOS and the microdrive operations. Any choice of disk drive, 5.25", 3.5", 3", single or double sided, 40 or 80 track, and any combination of up to four drives can be used. The interface automatically determines which type is being used and performs accordingly.

Provided that the disk size is the same, many disk drive files are compatible. ie. a file written on a 40T SS drive can be read on an 80T DS drive. And, an autoboot facility is also provided.

Most of the operating commands, except for a few special functions, are the same as the existing microdrive commands. All you have to do to use the disk drive is to use 'flp' instead of 'mdv' in SAVE/LOAD commands. If you are using the printer port or the RAM disk then use 'par' or 'ram' instead of 'mdv'.

These commands are mixable, so COPY-mdv1-file TO flp1-file will copy 'file' from microdrive 1 to disk drive 1. The names of these devices can be swapped around so that built in SAVE/LOAD functions may operate without program modification, ie. FLP-USE mdv will output everything to disk when a microdrive command is issued. Clever!

Practice

The interface is a neat unit designed in the same style as the QL, once plugged in it looks like part of the machine. On the left hand side of the case is the connector for the cable from the disk drive and at the back is the connector for the printer cable. The whole lot was assembled in less than ten minutes.

Although the printer interface is one of the lesser functions of the I/F it is very important as most printers on the market use this format, RS232 systems are available but usually cost extra to fit.

This worked a treat and a particularly useful function was the ability to specify the size of the print buffer. This can be to any size, memory permitting (and here the extra 128K was useful), and the result is that a lot of text can be held ready while the printer is operating, but the QL is left free for other purposes.

The documentation was a little sparse on information here, and a bit of trial and error was needed to get all the operations I wanted, but finally everything came together.

Bought separately, such an interface is priced at around £39.95.

RAM disk

Another extra, but again a very powerful one, is the ability to use a RAM disk. If a program is going to make many accesses to disk files during its operation then an area of memory (again the extra 128K was useful) can be used instead of the disk. This is created by giving it a number from 1 to 8 and specifying its size in sectors. All the disk information is transferred, altered, deleted or created in this area until the program is finished, at which point the whole lot can then be transferred back to disk.

This has two main advantages, first an incredible increase in speed, access is virtually instantaneous, and secondly less disk use means less wear and less chance of corrupted files. A little more sophisticated knowledge is required to use this and it's an option that is probably only really usable by accomplished programmers. Of course, it's a valuable facility for commercial software houses to use in their programs.

Disk drive

The amazing versatility in the tolerance of drive types is handy if you already own a drive. The only thing that is not too clearly stated is that the drive MUST have its own power supply.

Using this was as easy as the manual led me to believe, the increase in speed and reliability was comforting. Although we have not tested it for a long period of time, I have been using TR's Spectrum interface for a year now and it has not failed once, therefore I have every reason to trust in the reliability of their QL interface.

In fact it was a joy to use, and if this had been used instead of microdrives when the QL was first launched, Sinclair Research would have had a hard time keeping up with the orders. Now that it is here it should make a great difference to the machine's potential.

I hope software houses will make use of the facilities provided when designing their programs, and all users should consider this unit as being an important purchase. At the back of the manual is a wealth of technical information which hard/software experts will find invaluable. I may even abandon the Spectrum and Tasword II now that I have a reliable medium for storing text on the QL.

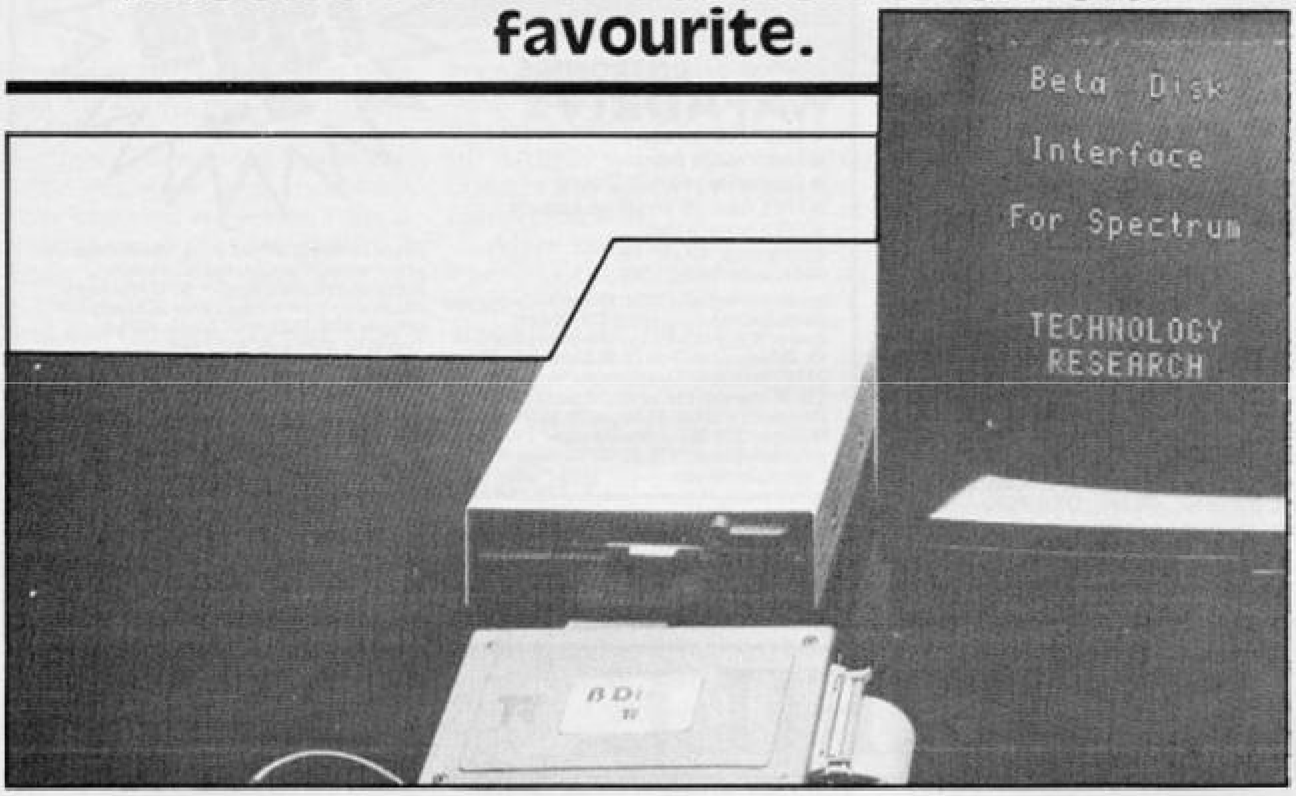
The QL plus the Delta Disk drive means business!

Delta Disk Interface £129.50: with 64K RAM extension, £199.00: with 128K extension £249.50. All prices are inclusive of VAT, but add £4.00 for p&p.

Technology Research Ltd, Unit 18, Central Trading Estate, Staines, Middlesex TW18 4XE. ■

The Spectrum Beta Plus Disk Drive

A look at a new version of an old favourite.



Technology Research Ltd. presented us with their new Beta Plus with an air of pride. Outwardly it looked the same as their old interface, a large box which lies flat behind the Spectrum, but with the addition of two small buttons. However the capabilities are greatly improved. To briefly recap on the original specifications for readers who missed our review is probably a good idea.

The Interface connects to the port at the rear of the Spectrum and has a through port on the back to connect further peripherals, it is fully compatible with the ZX Interfaces I and II, and has a power supply socket — using the original Spectrum PSU — on the left and the disk connector on the right.

Up to four drives can be connected and these may be any type, 40/80 track, single or double sided, and in any combination. They must however be double density drives and have their own PSUs.

A new feature of the interface is that the system checks the number and type of drives connected and operates them automatically when performing such operations as Format. Very clever. All the standard SAVE/LOAD operations are provided although VERIFY is not. After continual usage over a twelve month period with the first version, I have never had a failed save.

The original version did not provide sequential loading and saving a 255 byte sector of code was included. The new version is greatly improved by the addition of both sequential and random access file compatibility, the latter of which speeds up file operations. This was one of my few criticisms of the original.

The interface takes a mere 128 bytes of memory for its own use and is easy to use even if you only have a minimum of programming knowledge. The manual is a perfect example of how to write in a simple, clear manner yet contains all the information any user could require. Converting such programs as Tasword II or Masterfile, two of the most likely programs to be used with a disk system is very simple and takes only around ten minutes, and four of those are loading the original from tape!

The interface has proved to be compatible with all the disks used with its predecessor.

Extras

Especially interesting to those such as myself who enjoy playing games as well as using it for business is the new 'Magic Button'. This performs the same operation as the Miracle Microdriver does for Microdrives, in that the whole of the Spectrum's RAM memory is dumped to disk by pressing the small button at the rear of the interface.

This means that ANY program can be copied to disk, at least, I tried it with some twenty games which included those with different fast loaders and protective code, and all saved without problem.

There is a very slight problem in that this operation uses the first tracks on a disk, overwriting anything already there. This means that having used this operation you have to rename the file (all files so created are initially given a space as a filename) and then, using the utility provided on disk, transfer it to another disk. In this way I found you can store three programs on a 40 track single sided disk plus a main 'auto boot' menu program.

Oh yes, on power up or on pressing the little reset button on the side of the interface you now have the ability to automatically load and run a program called BOOT on the disk. If there is not such a program present then the system gives a disk error report and you can continue as you wish.

One thing that worried me at first was that, because of the automatic sensing of the drives, the drive is activated and the light remains on if no disk is in it on power up. Apart from giving me a moment of concern and a frantic call to TRL who assured me this was correct, this has no effect and, on inserting a disk, the drive whirrs and settles down. No ill effects to the disks have been experienced.

The updated utilities disk provides more powerful programs to backup, catalogue and copy files and each is simple to use, and a valuable bonus. Owners of the original TRL disk interface need not feel left out as they can update it for £23.95 inc. VAT & P&P, a reasonable price and well worth having done.

The interface is on sale direct from TRL at £111.25 inc. VAT and P&P.

Opinion

This is now THE disk drive interface for the Spectrum. I know of no other which provides such a comprehensive range of operations and facilities.

It's nearest competitor is the excellent Discovery all in one unit from Opus which, although lacking some of the TRL features has a few different ones to offer, such as printer, joystick and video ports. However TRL gave me the distinct impression that they have not finished with theirs yet . . .

Technology Research Ltd.,
Unit 18,
Central Trading Estate,
Staines.

Across the Pond

by Mark L. Fendrick

What is it about computers in general, and Sinclair computers in particular, that draws many of us so strongly to sit in front of a TV or monitor screen for hours at a time? The answer to that would have to be as varied as the people who own them, and it is in that variety that the answer lies. Tonight, major league baseball went on strike, and I find myself sitting at my desk looking to my T/S 2068 to fill the void. I find that I have to put my computer to many uses in the past few years, something that not too long ago would have been nothing more than science fiction. But Clive Sinclair was the one to change that for me, and since you are reading this column, probably for you as well. Once the pricetag for owning your own computer was in the thousands, but Uncle Clive (not yet Sir Clive) changed that with the ZX80 and ZX81, and those of us who had dreamed of having one could now afford it.

But, how many of us actually had a use in mind for our new computers when we first bought them? For that matter, to what use do we put them now? Has it become a paperweight for lack of a worthwhile occupation, or a workhorse used daily? It is with this question in mind that we open this month's column.

First and foremost at the present time are the wordprocessing abilities of my T/S 2068. When I first started writing a column, it was done on a conventional typewriter, which meant that once I typed a sentence or two, and decided to change it, I either had to resort to tipp-ex, or retype the entire page. With the wordprocessor, however, I am not committed to anything I type; even after it is printed. Since my best thoughts seem to always come the second time around, I cannot imagine being able to write this column any other way now.

Until recently, I made my changes, printed out a copy of the final result, and then consigned the column to the U.S. mail and hoped for the best. The past few columns, however, were sent from New York to Europe in just a matter of seconds via electronic mail. (Read on . . . the procedure for sending MSCRIPT files using MTERM follows later in this column.) From my T/S 2068 to an MCI print facility in Europe in less time than it took to read this sentence.

Comms

That brings us to another popular use for the Sinclair computers; telecommunication. With one of the two modems now on the market, you can turn your TS/ZX or T/S 2068 computer into a terminal to communicate with other computers via the telephone lines. In addition to the ability to send mail electronically, there are two other popular uses of a computer/modem combination. The first being the commercial services such as The Source, Compuserve, and Delphi, which provide a wide range of services. From games, to financial services, weather and news, for a fee (based on the number of minutes you are connected) there is something for almost anybody. One of the most popular features of these services is user-to-user communications. Realtime conversations with other computer owners all over the country are one of the byproducts of this new technology. Many Sinclair owners have found out about the weekly conference each Wednesday night (at 10.00pm) on CompuServe. By typing GO PCS-54 dozens of Sinclair computerists can get together to discuss a wide range of topics.

For those of you who don't care to spend the money involved with these services, there are the local bulletin boards maintained by individual computer enthusiasts. Many of them cater to special interests or particular computers. They consist mostly of a message base where callers can read and leave messages to each other, and some form of data base with such things as newsletters or public domain programs. One such bulletin board which caters to the Sinclair community can be reached at (718) 296-2229.

To be honest, though, when I first ordered my ZX81, I really did not have any specific use in mind, but soon thought of a few uses to justify the purchase. I was operating a photocopy business at the time, and decided that what I needed was an invoicing program. Also, the traditional home finance and checkbook uses seemed such a good idea, and I anticipated putting my ZX81 (with Memotech 64K add-on) to those uses. And not to ignore my wife's (less than) enthusiastic anticipation, I intended to transfer her gradebook to this new marvel.

In the months that followed, I retreated to the bedroom (where the computer now held residence) and started the development of the necessary software. Soon my business records, home finances and checkbook had entered the computer age, and the gradebook program grew to a full class management package. When I saw the lack of quality software of the market in those early days, I sought publishers, and my own Teacher's Gradebook (Timex), Invoice Control and the Financial Record Keeper series (Heath Computer Services) were marketed.

But how about you? What do you do with your Sinclair computer? Drop me a line and let me know what your computer is doing these days.

MSCRIPT

As promised, here is the procedure for sending MSCRIPT files using MTERM and your Westridge modem. A few facts which are necessary for you to know before you begin are the address of the start of the MTERM buffer, and how the MTERM software knows that there is something actually in the buffer. The answer to the first question is simple, as the MTERM buffer always starts at address 26710. You will need to know this address in order to place your MSCRIPT file in the proper memory.

Once you have SAVEd your MSCRIPT file to cassette in the normal manner, note the length of the file (from the MSCRIPT command menu) and then remove the tape from the recorder and place it aside. Now take a fresh, blank tape and place that into the recorder. Turn your computer off and then back on, to clear the MSCRIPT program from its memory. In command mode, type in; SAVE "file" CODE 26710, filelength, where filelength = the length of the MSCRIPT file noted earlier. You may, as an alternative, substitute a number such as 20000 which will cover any file you would ever want to LOAD, and keep this tape for future use. It is not necessary to have anything to SAVE in memory as all we are interested in here is the creation of a dummy header. The reason for this is that MSCRIPT SAVEs its files with a non-standard header which cannot be reLOADed using the normal Sinclair LOAD command. This will create a header with the information concerning the address to which the file is to be reLOADed, as well as the file length. Press the ENTER key, and any other key to start the SAVE procedure, but as soon as the header (the short first part of the SAVE) is recorded, you may stop the tape, and press the CAPS SHIFT/BREAK combination to stop the procedure. Now rewind this tape and remove it from the recorder. Replace the tape containing the MSCRIPT file (do NOT enter the MSCRIPT software though), and type; LOAD "". As soon as the header has been read (you will not get any file name) stop the tape and manually rewind the tape one quarter turn of the hub. Now remove the tape and put in the tape with the dummy header. Type LOAD "file" CODE and press ENTER. As soon as the header is recognised and read, stop the recorder (DO NOT PRESS BREAK), put back the MSCRIPT file tape and press PLAY on your recorder. You will see the familiar reading pattern, which will probably end with an error message. Don't worry, nothing is wrong, and it should be expected. Stop the tape, and LOAD your MTERM program, but do not PRINT USR 54016.

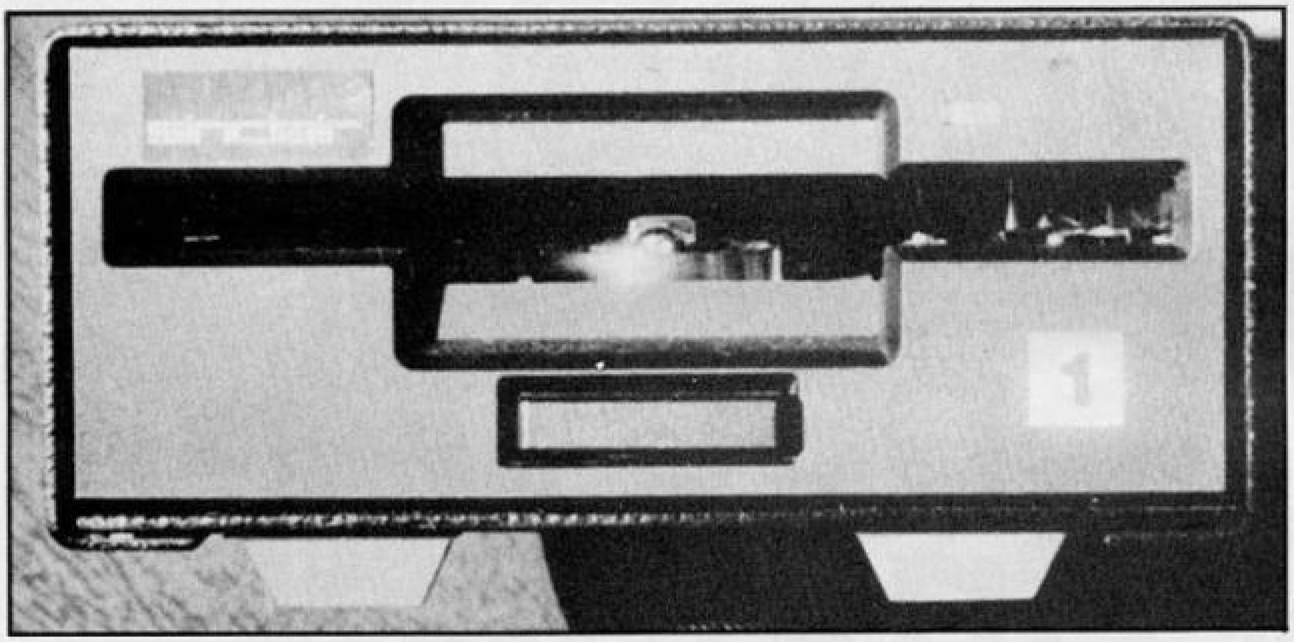
If at this point you were to go to the MTERM menu, the buffer would show that it is empty. This is because the end of the buffer is indicated by the system variable VARS. If you were to investigate, you would find that VARS is currently pointing to address 26710. POKE 23628, 200 to change that, and then activate MTERM. Now you will find that the program recognizes that your file is in the buffer. You can now transmit it in the normal manner.

Take my word for it, it is not really as complicated as it sounds, and once you do it once or twice you will see how easy it really is.


Try beating our SpecDrum!
Spec
Drum
Digital Drum System
for the Spectrum
£29.95
Exclusively available from
Boots
● 8 DIGITALLY RECORDED REAL DRUM SOUNDS
● SIMPLE TO USE — UP AND RUNNING IN ONLY MINUTES
● A TRUE DIGITAL DRUM MACHINE
● 'REAL TIME' OR ON-SCREEN PROGRAMMING
● COMPLETE WITH SOFTWARE
● EXTRA SOUNDS CAN BE LOADED FROM TAPE
● CREATIVE, EDUCATIONAL AND FUN
● THE MOST EXCITING PERIPHERAL EVER DEVELOPED
● DYNAMIC FILING SYSTEM — STORE OVER 1000 PROGRAMMED RYTHMS
● TAPE SYNC FACILITY
● SONGS CAN BE SAVED ON TAPE
● POLYPHONIC
● COMPREHENSIVE MANUAL
● JUST PLUG INTO YOUR HI FI
Cheetah
Marketing
1 WILLOWBROOK SCIENCE PARK, CRICKHOWELL ROAD, ST. MELLONS, CARDIFF
TELEPHONE: CARDIFF (0222) 777337 TELEX: 497455
(Export enquiries – contact Cheetah direct)

Micro Peripherals QL Disk System

We take a look at the Sinclair approved unit for disk storage on the QL.



It was a surprise to many when Sinclair Research officially endorsed the Micro Peripherals disk system, as it was the only one on the market at the time which did not contain software written by the QDOS author. However, when the system arrived on our desk for review we could see some of their reasons.

First Sight

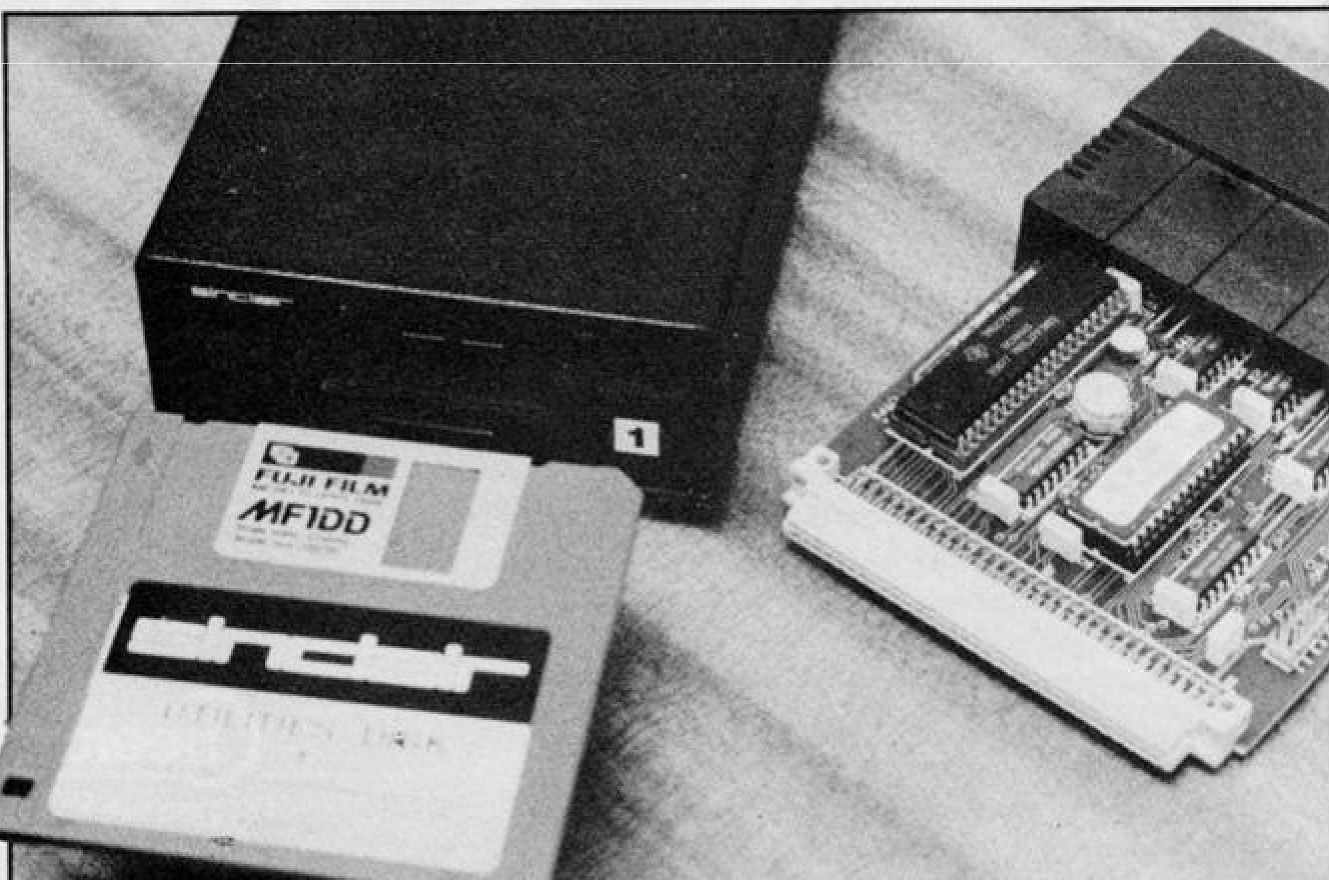
The interface looks much like any of the others on the market in that it fits into the port on the left of the QL, flush to the case and in the same style as the QL itself, becoming an integral part of the machine. The drives plug into a standard connector on the left.

It is possible to use the interface with drives other than their own providing the drive attached is 80 track, double density, single or double sided, has its own power supply and a maximum track step time of 6ms. However, probably due to the specific nature of the drive requirements, the marketing approach has been to offer one of two packages, an I/F plus one 3.5, 720KB (formatted), double sided, double density, 3ms Track step time, or an I/F plus two of these drives. The drives are small, neat black units with a PSU attached by a 6-pin Din plug to the centre of Drive 1. A ribbon connector cable runs from the drive to the I/F.

The second drive is different from Drive 1 in that it has a power lead ending in a 5-pin Din plug which connects to Drive 1 and a short ribbon cable which also connects to the rear of Drive 1. Both of these are permantly attached to Drive 2. These drives can sit side by side or on top of each other. The whole system looks very smart and attractive and compliments the QL in appearance.

The software consists of a set of detailed instruction sheets which fit into the QL manual. These are written in the same style as the manual, simple step-by-step initial instructions followed by in-depth details. The typeface is different but I found it easier on the eyes. Also included is a disk of utilities. These consist of:

REPOS_BAS, allowing modification of up to 9 windows.



BACKUP_X, a versatile file copying program.
EXCHANGE_X, a string search and replace of files.
DEL_FILES_X, to delete files.
EX_DIR_X, a detailed directory utility.
INSPECT_X, allowing examination and modification of specific files.
SPOOL_X, allowing transfer of files in ASCII form.
COPY_X, for single file transfers.
REDIRECT_X, moves files to various outputs such as a printer.
COLPRINT, allows printing from a mode 4 screen to a suitable colour printer.
D.A.M.E., a Disk and Memory Editor.

This latter program is a very powerful aid to manipulating files, working as it does at sector level. All these are excellent and provide a good range of valuable utilities for the experienced programmer.

In Use

Connecting to the QL is simplicity itself and no problems were experienced. There is an option to set the interface before fitting so that single or double sided drives are specified, track step time and whether or not the disk is called on power up instead of the microdrives for autoboot.

To use the disk systems FDK is typed rather than MDV, a software option exists to switch operating modes so that the disk operates on MDV instructions. DGET and DPUT give you the ability to read or write a 512 byte sector of the disk. VSET is useful as you can assign a three letter name of your own choice to the drives, I found this meant that existing software on disk using FLP could be used.

Using the system is reasonably straightforward although it was a good week before I felt confident enough to operate the system without the manual beside me. Having grasped the essentials the system produced no nasty surprises.

Conversion of the four Psion programs was accomplished with reasonable ease, although it took a few tries before I found that it worked best to copy the program, load the CONFIG_BAS program, flip the machine to use FDK in place of MDV then run the program and proceed as normal.

It seems strange that the four programs were not supplied ready for use on the disk, after all the originals were supplied with the machine and the unit is under the Sinclair banner. If this had been done it would have made it much more attractive for the owner or prospective buyer who only wanted to use the machine and not have to cope with this operation.

A variety of other software by Sinclair, Micro Power and Talent Systems also converted without problem.

Comments

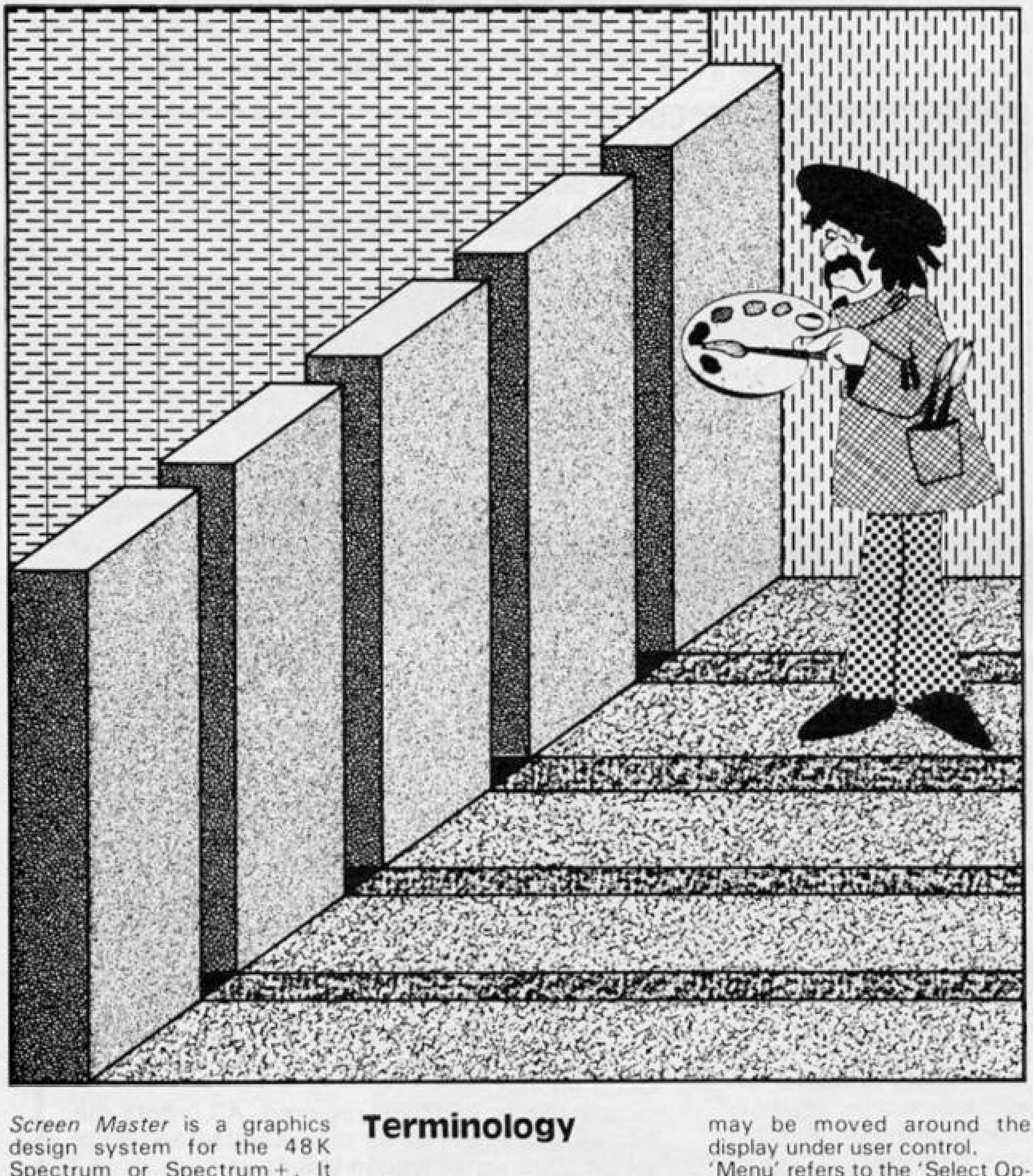
Essentially, this is an excellent unit. The biggest advantage of this system over others is that it is supplied as a package. This is bound to prove attractive to potential buyers as it can be purchased, connected and used immediately. It has proved reliable and a tremendous improvement over the microdrives, both in reliability of storage of files and in speed of access. Loading and saving of files is almost instantaneous, comparisons with similar units is not really worthwhile as, because the times are so small, differences are academic.

The disks themselves are more robust than the more common 5.25" type and therefore more reliable and less likely to suffer damage. Against the system is the fact that specific drives must be used and the most common 40 track 5.25" is unsuitable. This means that if you already possess one it is unusable. Also, the availability of 3.5" disks is not as widespread as the 5.25" kind, and they are much more expensive, (approx £5.00 compared with £2.00/£3.00).

A further problem which I was unaware of until I went to buy one, is that there is no 3.5" standard format. The one I bought would not fit into the drive! So, if you want an 'off the peg' system this one is ideal, but if you want equipment which allows more flexibility — such as additional memory and parrallel printer port — then you may well look at something like the Technology Research Interface, reviewed elsewhere in this issue. ■

Screen Master

For those of you wary of attempting our machine code graphics series, Matthew Probert presents a powerful screen artist program in BASIC.



Screen Master is a graphics design system for the 48K Spectrum or Spectrum+. It enables the user to quickly and simply design and save screens which may be used as title pages for programs during loading, or used within programs themselves by extracting the short machine code routine used by Screen Master to display a screen instantaneously.

Terminology

Within these instructions certain terms will be used which should be understood. These are as follows:

'Display' refers to the design on the TV screen.
'Editor or Block cursor' a two character square block which may be moved around the display under user control.
'Menu' refers to the 'Select Option state with the display showing the design being worked upon and the program waiting for the user to press a key.
'Pixel cursor' refers to a single pixel cursor which may be moved around the display under user control.
'Prompt' refers to a message displayed indicating that the user should press a key, ie 'Select option' is a prompt.
'Request' like 'prompt' but indicates that some data should be entered by the user.
'Screen' refers to a section of Ram holding the data of the design being worked upon.
'Text' refers to characters which may be displayed in 'Print' fashion.

Functions

Screen Master provides the user with eight different types of function for the display and manipulation of screen data. These functions are:

'Pixel functions' which operate basically upon single pixels.
'Block functions' which operate upon a block of data and tend to use the 'Editor cursor'.
'Screen functions' which operate upon screens of data.
'Tape functions' which use the cassette recorder.
'Attr functions' which change attributes.
'Text functions' which manipulate text.
'Brush functions' which provide various shading effects.
'Status functions' which alter or display the current status of the system.

Pixel functions

There are seven pixel functions; 'Freehand', 'Circle', 'Ellipse', 'Box', 'Move', and two others 'Move' and 'Erase' which are only accessed from 'Freehand'.

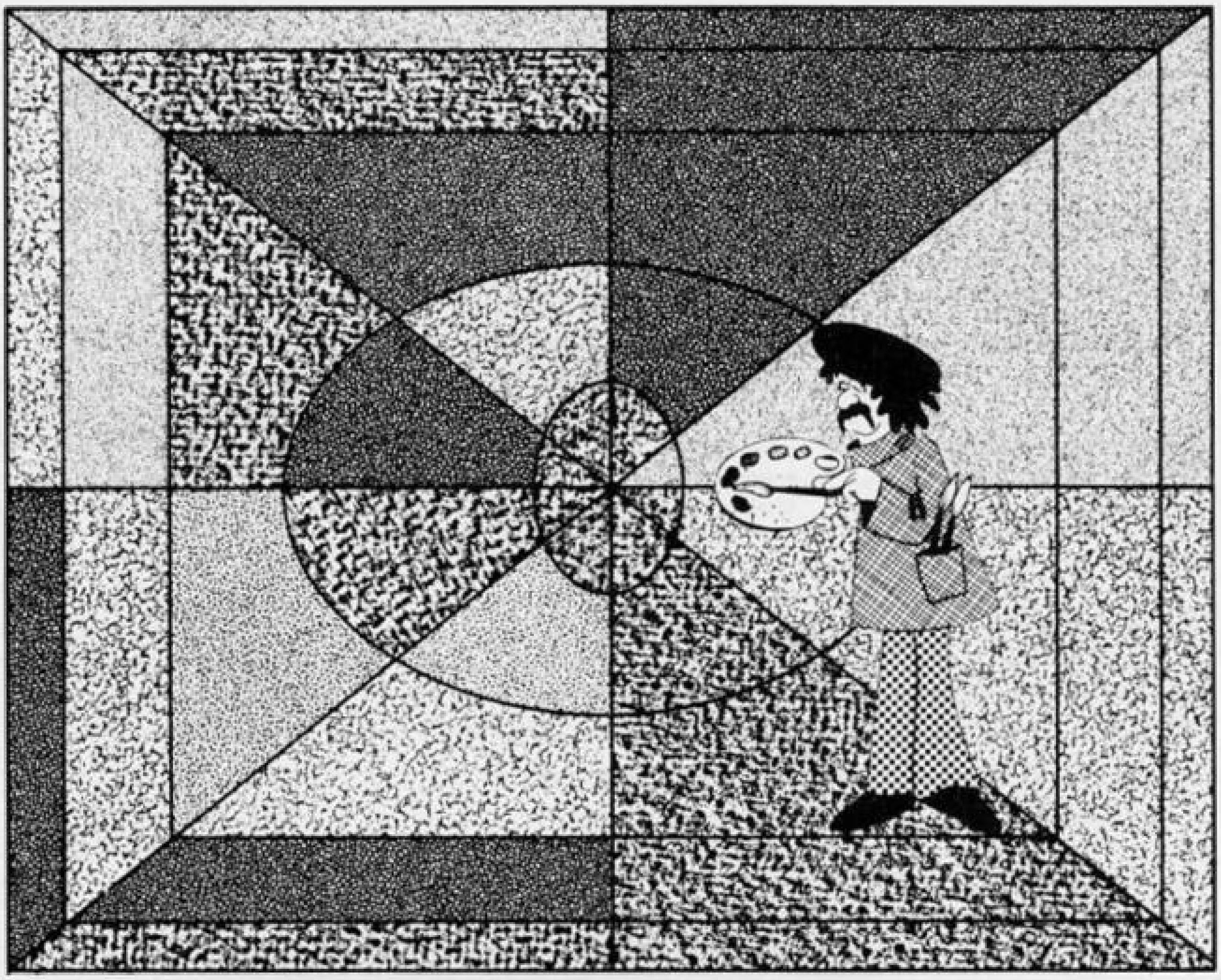
There are then three functions relating to 'Freehand', these plot points, move the pixel cursor and erase points respectively. From one function to another within Freehand the user may select either one of the other Freehand functions by pressing keys P, M and E respectively or exit back to the Menu by pressing Enter. The Freehand function is selected from the Menu by key F.

When in Freehand mode, the pixel cursor, like all other cursors may be moved around the display by the cursor keys or keys 5 to 8. During Freehand in all of its functions the coordinates of the pixel cursor, X,Y are displayed at the bottom of the display to aid the user.

The next three pixel functions, Circle, Ellipse and Box are selected by keys C, E and X respectively from the Menu. Each function requests an input from the user, for Circle this input is the radius of the circle in

pixels to be drawn with the coordinate X,Y as the centre of the circle. If 0 is entered as the radius the system will return to the Menu. Ellipse requests both the radius in the X and Y axis of the ellipse to be drawn. Entering 0 for the X radius will again return the user to the Menu, otherwise an ellipse will be drawn with the coordinates X,Y of the pixel cursor as its centre. Box works in a similar fashion, but draws a rectangle to the right and down from the pixel cursor, so that the coodinates X,Y form the top left corner of the shape. Entering a negative value in response to 'Length' will result in the rectangle being drawn to the left, and a negative value for height will result in it being drawn up.

Pixel function 'Move' as selected from the Menu by key L allows the user to move two cross lines around the screen at a rate of eight pixels per move. Upon exiting from this function the pixel cursor will be set to the coordinates at which the cross lines intersect.



Block functions

There are eight block functions, two of which use a special designer grid and these will be considered first.

'Sprite', selected by key S allows the user to design a two by two character block on a large grid a bit like a UDG generator. The cirsor on the grid may be moved in three ways, with the cursor keys or keys 5 to 8 as usual, and also with symbol shift and keys 5 to 8 together, in which case it will leave a trail of filled in squares behind it. Single squares may be filled with key 9 and deleted with key 0.

'Alter', selected by key A, is similar to Sprite, but picks up a two character square block of the screen, removes it from the display and transfers it to the grid ready for alterations to take place.

'Reflect' switches the data on either side of the Editor cursor either on top to bottom or left to right depending upon whether X or Y was entered in response to the request respectively. Reflect is selected by key R.

'Delete', selected by key DEL erases the data below the Editor cursor from the display and screen.

'Copy' is the same as Alter, but merely allows a copy of the data to be reproduced elsewhere without erasing the original. Copy is selected by key U.

'Switch', selected by key H, changes the attributes under the Editor cursor to the current values.

'Fill', selected by key N, fills in the area under the cursor as a solid block.

'Zap', selected by key Z changes the data under the cursor in a random way so as to give the impression that the data has been 'zapped'.

In all cases Enter should be used to exit from the move cursor routine, and Y or N pressed in response to the Okay? message. Pressing N reverts the display and screen back to how it was before the operation took place.

Screen functions

There are three screen functions, the first 'Cls' selected by pressing EDIT clears the display having first transferred the display into the current screen, this pressing EDIT twice consecutively will also clear the current screen as well as the display.

The second function, 'Down' selected by key W copies the screen to the display.

The last function 'Merge' selected by key M requests a screen number to be pressed and then overlays this screen on

```
Program 1
Type in this program and save on
tape, then type in the main prog
and save after this one using
LINE 1.

   1 DATA 33,0,64,17,87,228,1,19
2,26,237,176,201,33,87,228,17,0,
64,1,192,26,237,176,201
   2 FOR N=0 TO 23: READ a: POKE
 23296+n,a: NEXT N
   3 DATA 33,0,88,62,22,6,32,54,
5,35,16,251,61,254,0,32,244,201,
1,0,27,33,87,147,17,0,64,126,254
,0,40,1,18,35,19,11,120,177,200,
24,242
   4 FOR I=23320 TO 23360: READ
A: POKE I,A: NEXT I
   5 DATA 170,255,170,255,170,25
5,170,255,170,170,170,170,170,17
0,170,170,90,165,90,165,90,165,9
0,165
   6 DATA 255,0,255,0,255,0,255,
0,170,0,85,0,170,0,85,0,170,85,1
70,85,170,85,170,85,255,129,129,
129,129,129,129,255
   7 FOR N=0 TO 55: READ A: POKE
 USR "o"+N,A: NEXT N
  10 LOAD ""
```

top of the display in such a way that the new screen takes preference over the old, but leaves the attributes alone.

Tape functions

There are just two tape functions, 'Save' and 'Load', both of which operate using SCREEN$ and either save the display or load in the display and current screen. They are selected by Symbol Shift and either S or J respectively.

Attr functions

There are four Attr functions, the first, 'Ink' selected by key I changes the current ink colour, 'Paper' selected by key P changes the paper colour and 'Bright' selected by key B switches the brightness from on to off and from off to on automatically. All three of these work in 'temporary' fashion.

The last function, 'Chg' is selected by key O and changes the entire display and screen to the attribute value entered, it does not clear the display.

Text functions

There is one text function, selected by pressing key " it allows up to thirty two characters at a time to be displayed anywhere on the display and screen. The text may be graphics, selected by pressing the graphics key followed by entry of the required text, or ordinary characters.

The text may be displayed in inverse by pressing the INV Video key, or made to flash by pressing extend followed by CAPS shift and key 9 together. If you wish to exit the routine, pressing Q whilst in the Move cursor stage will return you to the Menu.

Brush functions

Six different brushes are provided, selected by keys 1 to 6 from the Menu. The brush function works like a cross between the text and Freehand functions described earlier.

Status functions

There are two status functions, 'Status' which is selected by pressing key K and displays the current ink, paper, bright, screen number and pixel cursor coordinates values, and 'Addr' which is selected by key T and allows the user to change screens between 1 and 4.

Technical details

The four screens are each 6848 bytes long and stored at addresses 58519, 51671, 44823, and 37975 respectively. The routine to copy the display into screen 'n' is at address 23296 with the screen address held in bytes 23300 and 23301. The routine is relocatable.

The routine to copy a screen to the display is at address 23308 with the screen address held in bytes 23309 and 23310. This routine is also relocatable.

The program uses seven UDG's, from O to U respectively. Ramtop is automatically reset to below screen 4 so that NEW does not affect the screens, in this way screens may be saved as Data, by; Pressing BREAK to break into Screen Master. Typing NEW to lose Screen Master, and either loading in your own program or saving the screen(s) as data.

Program 2. Main program.

```
   9 LET I=0: LET P=7: LET B=0:
LET A=0
  10 LET BL=0: LET X=128: LET Y=
88: LET X1=X: LET Y1=Y: LET Z=46
00: LET ST=23296: LET RT=23308:
LET PL=1: LET ZA=0
  15 LET G=5
  20 POKE 23658,8
  30 PAPER 8: INK 8: BRIGHT 8
  40 LET YN=5700
  55 LET DEL=20: LET XR=0: LET Y
R=0
  60 OPEN #G,"K"
 100 CLS
 120 INPUT "": PRINT #G;TAB 9;"S
ELECT  OPTION"
 130 GO SUB Z
 150 LET G$=INKEY$
 160 IF G$="F" THEN  GO SUB 1500
 170 IF G$="D" THEN  GO SUB 1200
 180 IF G$="C" THEN  GO SUB 1600
 190 IF G$="E" THEN  GO SUB 1700
 200 IF G$="A" THEN  GO SUB 2000
 210 IF G$="R" THEN  GO SUB 3000
 220 IF G$="I" THEN  GO SUB 3100
 230 IF G$="P" THEN  GO SUB 3200
 240 IF G$="B" THEN  GO SUB 3300
 250 IF G$="X" THEN  GO SUB 1800
 260 IF CODE G$=7 THEN  INPUT ""
: RANDOMIZE USR 23296: GO TO 10
 270 IF G$="W" THEN  RANDOMIZE U
SR 23308
 280 IF G$="""" THEN  GO SUB 370
0
 290 IF G$="NOT " THEN  GO SUB 3
900
 300 IF G$="-" THEN  GO SUB 4000
 310 IF G$>"0" AND G$<"7" THEN
GO SUB 4100
 320 IF G$="O" THEN  GO SUB 4300
 330 IF G$="T" THEN  GO SUB 3600
 340 IF CODE G$=12 THEN  GO SUB
4500
 350 IF G$="S" THEN  LET BL=1: G
O SUB 7000: LET BL=0
 360 IF G$="H" THEN  GO SUB 4700
 370 IF G$="U" THEN  GO SUB 5000
 380 IF G$="Z" THEN  LET ZA=1: G
O SUB 5300
 390 IF G$="M" THEN  GO SUB 5500
 400 IF G$="K" THEN  GO SUB 5600
 410 IF G$="L" THEN  GO SUB 5800
 420 IF G$="N" THEN  GO SUB 6200
 430 GO TO 110
1000 INPUT "": PRINT #G;TAB 3;"X
:";X1;TAB 10;"Y:";Y1;"";TAB 17;"
X1:";X;TAB 24;"Y1:";Y
1005 IF CODE INKEY$=13 THEN  GO
TO 1005
1016 PLOT  OVER 1;X1,Y1
1020 LET G$=INKEY$
1030 IF (G$>="5" AND G$<="8") OR
 (CODE G$>=8 AND CODE G$<=11) TH
EN  GO TO 1050
1035 IF CODE G$=13 THEN  RETURN
1040 GO TO 1020
1060 LET X1=X1-(G$="5" OR CODE G
```

```
$=8 AND X1>Ø)+(G$="8" OR CODE G$
=9 AND X1<255)
1070 LET Y1=Y1-(G$="6" OR CODE G
$=10 AND Y1>Ø)+(G$="7" OR CODE G
$=11 AND Y1<175)
1075 INPUT "": PRINT #G;TAB 3;"X
:";X1;TAB 10;"Y:";Y1;"";TAB 17;"
X1:";X;TAB 24;"Y:";Y1
1080 PLOT  OVER 1;PEEK 23677,PEE
K 23678
1085 PLOT  OVER 1;X1,Y1
1090 GO TO 1020
1105 INPUT "": PRINT #G;TAB 3;"X
:";X1;TAB 10;"Y:";Y1;TAB 17;"X1:
";X;TAB 24;"Y1:";Y
1110 PLOT  OVER 1;X,Y
1120 LET G$=INKEY$
1130 IF (G$>="5" AND G$<="8") OR
 (CODE G$>=8 AND CODE G$<=11) TH
EN  GO TO 1150
1135 IF CODE G$=13 THEN  RETURN
1140 GO TO 1120
1160 LET X=X-(G$="5" OR CODE G$=
8 AND X>Ø)+(G$="8" OR CODE G$=9
AND X<255)
1170 LET Y=Y-(G$="6" OR CODE G$=
10 AND Y>Ø)+(G$="7" OR CODE G$=1
1 AND Y<175)
1175 INPUT "": PRINT #G;TAB 3;"X
:";X1;TAB 10;"Y:";Y1;TAB 17;"X1:
";X;TAB 24;"Y1:";Y
1180 PLOT  OVER 1;PEEK 23677,PEE
K 23678
1185 PLOT  OVER 1;X,Y
1190 GO TO 1120
1200 INPUT TAB 8;"ANGLE OF ARC?
"; LINE Q$: IF Q$>="A" THEN  GO
TO 1200
1201 LET A=VAL Q$
1202 PRINT #G;TAB 14;"DRAW"
1203 PRINT  FLASH 1; OVER 1;AT 2
1-Y1/8,X1/8;" "
1204 PAUSE 100: PRINT  FLASH Ø;
OVER 1;AT 21-Y1/8,X1/8;" "
1205 IF CODE INKEY$=13 THEN  GO
TO 1205
1210 GO SUB 1000
1217 PRINT  FLASH 1; OVER 1;AT 2
1-Y/8,X/8;" "
1218 PAUSE 100: PRINT  FLASH Ø;
OVER 1;AT 21-Y/8,X/8;" "
1219 IF CODE INKEY$=13 THEN  GO
TO 1219
1220 GO SUB 1100
1230 DRAW  INK I; PAPER P; BRIGH
T B;X1-X,Y1-Y,A
1240 GO SUB YN
1260 IF INKEY$="Y" THEN  RANDOMI
ZE USR ST: RETURN
1270 PLOT  OVER 1;X,Y: DRAW  INK
 I; PAPER P; BRIGHT B; OVER 1;X1
-X,Y1-Y,A
1280 GO TO 1200
1500 INPUT "": PRINT #G;TAB 12;"
FREEHAND"
1505 IF INKEY$<>"" THEN  GO TO 1
505
1510 PLOT  OVER 1;X,Y
1520 LET G$=INKEY$
1525 IF G$="M" THEN  LET PL=Ø: I
NPUT "": PRINT #G;TAB 13;"MOVE":
 IF NOT POINT (X,Y) THEN  PLOT
OVER 1;X,Y
1526 IF G$="P" THEN  LET PL=1: I
NPUT "": PRINT #G;TAB 10;"FREEHA
ND"
1527 IF G$="E" THEN  LET PL=2: I
NPUT "": PRINT #G;TAB 13;"ERASE"
1530 IF (G$>="5" AND G$<="8") OR
 (CODE G$>=8 AND CODE G$<=11) TH
EN  GO TO 1550
1532 IF (CODE G$=13 AND (X=128 A
ND Y=88) OR PL=Ø) THEN  PLOT  OV
ER 1;PEEK 23677,PEEK 23678
1535 IF CODE G$=13 THEN  RETURN
1540 GO TO 1510
1550 IF PL=1 THEN  PLOT  INK I;
PAPER P; BRIGHT B;X,Y
1551 IF PL=Ø THEN  PLOT  OVER 1;
X,Y
1552 IF PL=2 AND POINT (X,Y)=1 T
HEN  PLOT  OVER 1;X,Y
1555 LET X=X-(G$="5" OR CODE G$=
8 AND X>Ø)+(G$="8" OR CODE G$=9
AND X<255)
1560 LET Y=Y-(G$="6" OR CODE G$=
10 AND Y>Ø)+(G$="7" OR CODE G$=1
1 AND Y<175)
1566 INPUT "": PRINT #G;TAB 10;"
X:";X;TAB 17;"Y:";Y
1567 IF PL=2 AND POINT (X,Y)=1 T
HEN  PLOT  OVER 1;X,Y
1570 GO TO 1510
1600 INPUT TAB 10;"RADIUS "; LIN
E Q$: IF Q$>="A" THEN  GO TO 160
Ø
1610 LET R=VAL Q$: IF NOT R THEN
 RETURN
1620 IF X+R>255 OR X-R<Ø OR Y+R>
175 OR Y-R<Ø THEN  RETURN
1630 CIRCLE  INK I; PAPER P; BRI
GHT B;X,Y,R
1640 GO SUB YN
1660 IF INKEY$="Y" THEN  RANDOMI
```

```
ZE USR ST: RETURN
167Ø RANDOMIZE USR RT
168Ø GO TO 16ØØ
17ØØ INPUT TAB 9;"X RADIUS "; LI
NE Q$: IF Q$>="A" THEN  GO TO 17
ØØ
171Ø LET XR=VAL Q$: IF NOT XR TH
EN  RETURN
172Ø INPUT TAB 9;"Y RADIUS "; LI
NE Q$: IF Q$>="A" THEN  GO TO 17
2Ø
1725 LET YR=VAL Q$
173Ø IF X+XR>255 OR X-XR<Ø OR Y+
YR>175 OR Y-YR<Ø THEN  RETURN
174Ø LET NA=XR*YR: IF NA>3ØØØ TH
EN  LET NA=1: GO TO 175Ø
1745 LET NA=2
175Ø FOR N=1 TO 72Ø/NA
176Ø PLOT  INK I; BRIGHT B; PAPE
R P;X+XR*COS (N/(36Ø/NA)*PI),Y+Y
R*SIN (N/(36Ø/NA)*PI)
177Ø NEXT N
178Ø GO SUB YN
1783 IF INKEY$="Y" THEN  RANDOMI
ZE USR ST: RETURN
1784 RANDOMIZE USR RT: GO TO 17Ø
Ø
18ØØ INPUT "LENGTH "; LINE Q$: I
F Q$>="A" THEN  GO TO 18ØØ
181Ø LET L=VAL Q$: IF NOT L THEN
  RETURN
182Ø INPUT "HEIGHT "; LINE Q$: I
F Q$>="A" THEN  GO TO 182Ø
1825 LET H=VAL Q$
183Ø IF X+L>255 OR Y-H<Ø THEN  R
ETURN
184Ø PLOT  INK I; PAPER P; BRIGH
T B;X,Y: DRAW  INK I; PAPER P; B
RIGHT B;L,Ø: DRAW  INK I; PAPER
P; BRIGHT B;Ø,-H: DRAW  INK I; B
RIGHT B; PAPER P;-L,Ø: DRAW  INK
 I; BRIGHT B; PAPER P;Ø,H
185Ø RETURN
2ØØØ POKE 23728,X: POKE 23729,Y
2ØØ2 GO SUB 34ØØ
2ØØ3 LET AT=22528+C+32*R
2ØØ5 DIM Q(4)
2Ø1Ø LET Q(1)=ATTR (R,C): LET Q(
2)=ATTR (R,C+1): LET Q(3)=ATTR (
R+1,C): LET Q(4)=ATTR (R+1,C+1)
2Ø3Ø LET X1=C*8: LET Y1=175-R*8
2Ø4Ø DIM B$(16,16)
2Ø8Ø INPUT "": PRINT #G;TAB 1Ø;"
PLEASE WAIT"
2Ø9Ø IF BL=1 THEN  FOR N=1 TO 16
: LET B$(N)="□□□□□□□□□□□□□□□□":
NEXT N: GO TO 217Ø
```

```
21ØØ FOR Y=Ø TO 15
21Ø5 LET Y1=(175-R*8)-Y
211Ø FOR X=Ø TO 15
212Ø LET X1=C*8+X
213Ø IF POINT (X1,Y1)=1 THEN  LE
T B$(X+1,Y+1)="■": PLOT  OVER 1;
X1,Y1: GO TO 215Ø
214Ø IF POINT (X1,Y1)=Ø THEN  LE
T B$(X+1,Y+1)="□"
215Ø NEXT X
216Ø NEXT Y
2162 IF ZA THEN  GO TO 53Ø5
2164 RANDOMIZE USR ST
2165 IF YR THEN  GO SUB 28ØØ: GO
 TO 25ØØ
2166 IF XR THEN  GO SUB 27ØØ: GO
 TO 25ØØ
217Ø CLS
218Ø FOR N=1 TO 16
219Ø FOR M=1 TO 16
22ØØ PRINT AT N,M;B$(M,N)
221Ø NEXT M
222Ø NEXT N
2235 PRINT AT 1,19;"MOVE WITH";A
T 3,19;"CURSOR KEYS";AT 5,19;"FI
LL WITH 9";AT 7,19;"DEL WITH Ø";
AT 9,19;"ENTER TO END"
224Ø LET A=1: LET AA=1
225Ø PRINT  OVER 1; FLASH 1;AT A
,AA;"X"
226Ø LET G$=INKEY$
227Ø IF G$="" THEN  GO TO 226Ø
2285 BEEP .Ø1,2Ø
229Ø IF G$="9" OR G$="(" OR G$="
'" OR G$="&" OR G$="%" THEN  PRI
NT AT A,AA;"■": LET B$(AA,A)="▩"
2295 PRINT AT A,AA;B$(AA,A)
23ØØ IF G$="Ø" THEN  PRINT AT A,
AA;"□": LET B$(AA,A)="□"
231Ø LET AA=AA+(G$="8" OR CODE G
$=9 OR G$="(" AND AA<16)-(G$="5"
 OR CODE G$=8 OR G$="%" AND AA>1
)
232Ø LET A=A+(G$="6" OR CODE G$=
1Ø OR G$="&" AND A<16)-(G$="7" O
R CODE G$=11 OR G$="'" AND A>1)
2325 IF AA>16 THEN  LET AA=16
2326 IF A>16 THEN  LET A=16
2327 IF A<1 THEN  LET A=1
2328 IF AA<1 THEN  LET AA=1
233Ø IF CODE G$=13 THEN  GO TO 2
5ØØ
234Ø GO TO 225Ø
25Ø6 CLS : RANDOMIZE USR RT
25Ø7 INPUT "": PRINT #G;TAB 9;"S
AME PLACE?"
25Ø8 IF INKEY$<>"Y" AND INKEY$<>
```

```
"N" THEN  GO TO 2508
2509 IF INKEY$="Y" THEN  LET C1=
C: LET R1=R: GO TO 2515
2510 GO SUB 3400: LET R1=R: LET
C1=C
2511 LET AT=22528+C1+32*R1
2512 FOR N=0 TO 15
2513 LET Q(1)=ATTR (R1,C1): LET
Q(2)=ATTR (R1,C1+1): LET Q(3)=AT
TR (R1+1,C1): LET Q(4)=ATTR (R1+
1,C1+1)
2515 INPUT "": PRINT #G;TAB 10;"
PLEASE WAIT"
2516 FOR N=0 TO 15
2520 FOR M=0 TO 15
2530 IF B$(M+1,N+1)="■" THEN  PL
OT C1*8+M,(175-R1*8)-N
2540 NEXT M
2550 NEXT N
2560 POKE AT,Q(1): POKE AT+1,Q(2
): POKE AT+32,Q(3): POKE AT+33,Q
(4)
2562 IF ZA THEN  GO TO 5360
2563 RANDOMIZE USR ST
2564 LET X=PEEK 23728: LET Y=PEE
K 23729
2565 RETURN
2730 FOR M=1 TO 16
2740 FOR N=1 TO 7
2745 LET T$=B$(8+N,M)
2750 LET B$(8+N,M)=B$(N,M)
2755 LET B$(N,M)=T$
2760 NEXT N
2770 NEXT M
2790 RETURN
2830 FOR M=1 TO 7
2840 FOR N=1 TO 16
2845 LET T$=B$(N,8+M)
2850 LET B$(N,8+M)=B$(N,M)
2855 LET B$(N,M)=T$
2860 NEXT N
2870 NEXT M
2880 RETURN
3000 INPUT "    REFLECT ABOUT X
OR Y"; LINE Q$
3010 IF Q$<>"X" AND Q$<>"Y" THEN
 GO TO 3000
3020 LET XR=Q$="X"
3030 LET YR=Q$="Y"
3040 GO TO 2000
3100 INPUT "": PRINT #G;"  PRESS
 0 TO 9 FOR  INK COLOUR"
3110 IF INKEY$<>"" THEN  GO TO 3
110
3120 IF INKEY$<"0" OR INKEY$>"9"
 THEN  GO TO 3120
3130 LET I=VAL INKEY$
3140 RETURN
3200 INPUT "": PRINT #G;" PRESS
0 TO 9 FOR  PAPER COLOUR"
3210 IF INKEY$<>"" THEN  GO TO 3
210
3220 IF INKEY$<"0" OR INKEY$>"9"
 THEN  GO TO 3220
3230 LET P=VAL INKEY$
3240 RETURN
3300 LET B=B=0
3310 RETURN
3400 LET R=10: LET C=15
3410 PRINT  OVER 1;AT R,C;"  ";A
T R+1,C;"  "
3420 INPUT "": PRINT #G;TAB 4;"M
OVE SQUARE TO POSITION"
3430 LET G$=INKEY$
3435 IF CODE G$=13 THEN  PRINT
OVER 1;AT R,C;"  ";AT R+1,C;"  "
: RETURN
3437 IF G$="Q" THEN  RETURN
3440 IF G$="" THEN  GO TO 3430
3450 LET R1=R+(G$="6" OR CODE G$
=10 AND R<20)-(G$="7" OR CODE G$
=11 AND R>0)
3460 LET C1=C+(G$="8" OR CODE G$
=9 AND C<31)-(G$="5" OR CODE G$=
8 AND C>0)
3470 PRINT  OVER 1;AT R,C;"  ";A
T R+1,C;"  "
3480 LET R=R1: LET C=C1
3490 PRINT  OVER 1;AT R,C;"  ";A
T R+1,C;"  "
3500 GO TO 3430
3610 INPUT "": PRINT #G;TAB 8;"S
ELECT 1 TO 4"
3620 IF INKEY$<>"" THEN  GO TO 3
620
3630 IF INKEY$<"1" OR INKEY$>"4"
 THEN  GO TO 3630
3635 LET G$=INKEY$
3640 LET S=65367-VAL G$*6848
3650 POKE 23300,S-256*INT (S/256
): POKE 23301,INT (S/256): POKE
23309,PEEK 23300: POKE 23310,PEE
K 23301
3660 INPUT "": PRINT #G;TAB 10;"
SCREEN=";VAL G$
3670 GO SUB Z
3690 RETURN
3710 INPUT "TEXT "; LINE T$
3715 LET R=0: LET C=0: PRINT  IN
K 8; OVER 1; FLASH 1;AT R,C;"■"
3720 PRINT #5;"MOVE SQUARE TO RE
QUIRED POSITION"
3725 IF INKEY$<>"" THEN  GO TO 3
725
```

```
3730 LET G$=INKEY$: IF G$="" THE
N GO TO 3730
3740 IF CODE G$=13 THEN GO TO 3
800
3750 PRINT OVER 1; FLASH 0;AT R
,C;"■"
3760 LET R=R+(G$="6" OR CODE G$=
10 AND R<21)-(G$="7" OR CODE G$=
11 AND R>0)
3770 LET C=C+(G$="8" OR CODE G$=
9 AND C<31)-(G$="5" OR CODE G$=8
 AND C>0)
3780 PRINT OVER 1; FLASH 1;AT R
,C;"■"
3790 GO TO 3730
3800 PRINT INK I; BRIGHT B; PAP
ER P;AT R,C;T$
3810 GO SUB YN
3830 IF INKEY$="Y" THEN INPUT "
": RANDOMIZE USR ST: RETURN
3835 INPUT ""
3840 RANDOMIZE USR RT
3845 PRINT OVER 1;AT R,C;"■"
3850 GO TO 3720
3905 DIM S$(10)
3910 INPUT "ENTER TITLE FOR SAVI
NG"; LINE S$
3915 RANDOMIZE USR ST: BORDER P:
 CLS : RANDOMIZE USR RT
3920 SAVE S$SCREEN$
3930 RETURN
4010 INPUT "ENTER TITLE FOR LOAD
ING"; LINE S$
4015 IF S$="          " THEN LO
AD ""SCREEN$ : GO TO 4030
4020 LOAD S$SCREEN$
4030 RANDOMIZE USR ST: RETURN
4105 LET CH=164-VAL G$
4110 LET R=0: LET C=0
4120 PRINT OVER 1;AT R,C;CHR$ C
H
4130 INPUT "": PRINT #G;"MOVE BR
USH TO REQUIRED POSITION "
4140 LET G$=INKEY$: IF G$="" THE
N GO TO 4140
4145 PRINT OVER 1;AT R,C;CHR$ C
H
4150 LET R=R+(G$="6" OR CODE G$=
10 AND R<21)-(G$="7" OR CODE G$=
11 AND R>0)
4160 LET C=C+(G$="8" OR CODE G$=
9 AND C<31)-(G$="5" OR CODE G$=8
 AND C>0)
4170 PRINT OVER 1;AT R,C;CHR$ C
H
4180 IF CODE G$<>13 THEN GO TO
4140
4190 INPUT "": PRINT #G;TAB 7;"M
OVE BRUSH TO PAINT"
4195 GO SUB Z
4200 LET G$=INKEY$: IF G$="" THE
N GO TO 4200
4220 LET R=R+(G$="6" OR CODE G$=
10 AND R<21)-(G$="7" OR CODE G$=
11 AND R>0)
4230 LET C=C+(G$="8" OR CODE G$=
9 AND C<31)-(G$="5" OR CODE G$=8
 AND C>0)
4232 PRINT INK I; PAPER P; BRIG
HT B;AT R,C;CHR$ CH
4235 INPUT "": PRINT #G;TAB 8;"R
OW ";R;TAB 18;"COLUMN ";C
4240 IF CODE G$=13 THEN INPUT "
": RANDOMIZE USR ST: RETURN
4250 GO TO 4200
4310 INPUT "ENTER REQUIRED ATTR
VALUE";A
4320 IF A>255 OR A<0 THEN GO TO
 4310
4330 POKE 23328,A
4340 RANDOMIZE USR 23320
4345 IF A>127 THEN LET A=A-128
4346 IF A>63 THEN LET A=A-64: L
ET B=1
4350 LET P=INT (A/8): LET I=A-(I
NT (A/8)*8)
4360 RETURN
4510 GO SUB 3400
4520 PRINT AT R,C;"  ";AT R+1,C;
"  "
4530 GO SUB YN
4540 IF INKEY$="Y" THEN INPUT "
": RANDOMIZE USR ST: RETURN
4550 GO SUB 3410
4560 GO TO 4520
4600 IF INKEY$<>"" THEN GO TO 4
600
4610 IF INKEY$="" THEN GO TO 46
10
4620 RETURN
4700 GO SUB 3400
4710 PRINT PAPER P; INK I; OVER
 1; BRIGHT B;AT R,C;"  ";AT R+1,
C;"  "
4720 RETURN
5010 GO SUB 3400
5020 DIM B$(16,16)
5030 INPUT "": PRINT #G;TAB 10;"
PLEASE WAIT"
5040 FOR Y=0 TO 15
5050 LET Y1=(175-R*8)-Y
5060 FOR X=0 TO 15
5070 LET X1=C*8+X
5080 IF POINT (X1,Y1) THEN LET
```

```
B$(X+1,Y+1)="■"
5090 IF NOT POINT (X1,Y1) THEN
LET B$(X+1,Y+1)="□"
5100 NEXT X
5110 NEXT Y
5120 INPUT "": GO SUB 3400
5130 LET C1=C: LET R1=R
5140 INPUT "": PRINT #G;TAB 10;"
PLEASE WAIT"
5150 FOR N=0 TO 15
5160 FOR M=0 TO 15
5170 IF B$(M+1,N+1)="■" THEN  PL
OT  INK I; PAPER P; BRIGHT B;C1*
8+M,(175-R1*8)-N
5180 NEXT M
5190 NEXT N
5200 RANDOMIZE USR ST
5210 RETURN
5300 GO TO 2000
5302 POKE 23670,PEEK 23673: POKE
 23671,PEEK 23672
5305 FOR N=0 TO 20+INT (RND*40)
5310 LET R2=INT (RND*16)+1: LET
C2=INT (RND*16)+1
5320 IF B$(R2,C2)="■" THEN  LET
B$(R2,C2)="□": GO TO 5340
5330 LET B$(R2,C2)="■"
5340 NEXT N
5350 GO TO 2507
5360 GO SUB YN
5375 LET ZA=0
5377 LET X=PEEK 23728: LET Y=PEE
K 23729
5380 IF INKEY$="Y" THEN  RANDOMI
ZE USR ST: RETURN
5390 RANDOMIZE USR RT
5400 RETURN
5500 INPUT "": PRINT #G;TAB 8;"S
ELECT 1 TO 4"
5510 IF INKEY$<>"" THEN  GO TO 3
620
5520 IF INKEY$<"1" OR INKEY$>"4"
 THEN  GO TO 5520
5525 LET G$=INKEY$
5530 LET S=65367-VAL G$*6912
5540 POKE 23342,S-256*INT (S/256
): POKE 23343,INT (S/256)
5550 RANDOMIZE USR 23338
5580 RETURN
5600 CLS
5610 PRINT AT 5,13;"STATE": PRIN
T ''TAB 11;"INK.....";I: PRINT
TAB 11;"PAPER...";P: PRINT 'TAB
11;"BRIGHT..";: PRINT ("ON" AND
B=1)+("OFF" AND B=0)
5620 PRINT 'TAB 11;"SCREEN #";
5630 LET S=PEEK 23300+256*PEEK 2
3301: LET S=65367-S: LET S=S/691
2
5640 PRINT S
5645 PRINT 'TAB 11;"X,Y.....";X;
",";Y
5650 GO SUB Z: CLS : RANDOMIZE U
SR RT: RETURN
5700 INPUT "": PRINT #G;TAB 13;"
OKAY?"
5710 IF INKEY$<>"Y" AND INKEY$<>
"N" THEN  GO TO 5710
5720 RETURN
5800 INPUT "": PRINT #G;"   MOVE
 CROSS LINES TO POSITION"
5860 LET X=128: LET Y=88
5870 GO SUB 6100
5800 INPUT "": PRINT #G;"   MOVE
 CROSS LINES TO POSITION"
5860 LET X=128: LET Y=88
5870 GO SUB 6100
5880 LET G$=INKEY$: IF G$="" THE
N  GO TO 5880
5890 IF G$>"8" THEN  GO TO 5880
5895 IF CODE G$=13 THEN  GO TO 6
130
5900 GO SUB 6100
6010 LET X=X+(8 AND (G$="8" OR C
ODE G$=9 AND X<248))-(8 AND (G$=
"5" OR CODE G$=8 AND X>7))
6020 LET Y=Y+(8 AND (G$="7" OR C
ODE G$=11 AND Y<167))-(8 AND (G$
="6" OR CODE G$=10 AND Y>7))
6040 GO TO 5870
6100 PLOT  OVER 1;X,175: DRAW  O
VER 1;0,-175: PLOT  OVER 1;0,Y:
DRAW  OVER 1;255,0
6120 RETURN
6130 GO SUB YN
6140 IF INKEY$="N" THEN  INPUT "
": PRINT #G;"   MOVE CROSS LINES
 TO POSITION": GO TO 5880
6150 RANDOMIZE USR RT
6160 RETURN
6200 GO SUB 3400
6205 IF G$="Q" THEN  RANDOMIZE U
SR RT: RETURN
6210 PRINT  INK I; PAPER P; BRIG
HT B;AT R,C;"■■";AT R+1,C;"■■"
6220 GO SUB YN
6230 IF INKEY$="N" THEN  RANDOMI
ZE USR RT: GO TO 6200
6235 RANDOMIZE USR ST
6240 INPUT "": PRINT #G;TAB 10;"
 CONTINUE ?"
6250 GO SUB 5710
6260 IF INKEY$="N" THEN  RETURN
6270 GO TO 6200
```

Wise Moves

Michael Spencer shows us some ways of getting things moving around the screen.

Animation is the process that puts life into computer games, but has far wider uses. The ability of a computer to create, move and display visual information has only reached the home market fairly recently,and is becoming a very popular and interesting application for users at all levels.

Computer graphics is a term that means different things to different people, but generally covers images that are computer generated, modified, stored and displayed but need not necessarily be animated. To a businessman, 'graphics' means histograms, graphs and other methods of presenting financial information visually; to a railway signalman the mimic display showing train moves is a form of computer graphic; to a student pilot on an aircraft simulator the realistic images are computer generated. Computer aided design perhaps involves the most complicated imagery of all, think about car design as an example. A good deal of computer graphics is merely static images that can be displayed when required, animation is the art of making those images appear to move in such a way that the impression of a lifelike object is created. Animation is an art that predates the computer revolution by quite a long time and its application is a practised skill. Cartoons are a much loved example of film animation and such films have given pleasure to millions, but few of these millions have ever given much thought to the sheer effort involved in producing those cartoons. In one second of film there will be twenty four frames and each frame is an individual picture showing a stage in the motion of the animated object. It doesn't take much mental arithmetic to work out how many individual pictures there are likely to be in a ten minute cartoon feature.

Cartoons are an example of drawn animation that uses the skill of an animator to draw a sequence of individual pictures which, when filmed, give the impression of motion. There are other techniques of animation-model, cutout and rostrum (controlled camera) animation are examples.These techniques are more suitable to the use of computers to aid such activity, drawn animation is more directly applicable to using computers to replace manual techniques and thid is the main topic of discussion here.

Animators in the film industry are skilled people and the success of the work depends on their grasp of two factors-timing and realism. Correct timing can convey the impression of animation well and that timing ought to reflect the behaviour in the real world of the object being animated. Realism ensures that the object and its behaviour reflect the laws of motion and the presence of gravity. These techniques have been built up over many years and it is only their application in the use of computers that is new — the computer screen is only another medium on which to display images (it may be called electronic paper). The computer may also be a means to an end, computers as animators' tools are having an impact on the film industry. Remember the Walt Disney film TRON about a computer programmer and his battle against a wayward super computer? A good deal of that film was made by using computer animation and still graphics — a novel approach that created a stunning impact. Computers can take the tedium out of the process of producing images for animation — a single image can be manipulated to a new position, size or shape by applying mathmatical routines. Taken to an extreme it may be possible to produce a complete film in a few years time, and indeed a Hollywood film company is using a CRAY super computer to cope with the fantastic number of calculations involved in such processes. This machine made its debut code cracking in American for the Government and has since made contributions in many other fields.

Animation itself involves three basic processes — preparation of the image, manipulation and display. It is the manipulation that is at the heart of the technique of animation. The computer offers the electronic paper for us to exploit and the characteristics of that paper determine what sort of animation is possible. There are three sorts of graphics that can be created on most home computers — block, pixel and line graphics. Block graphics involve the use of character sized shapes (see the top row of the Spectrum keyboard) which may be used individually or as part of larger shapes which can be moved around the screen. The program in figure one creates such a shape and moves it around. The program is reasonably effective and shows some of the basics of simple animation. An image drawn on the screen is erased and a new image created in a different place which gives the aperance of motion. A variation of the technique would be to surround the image with blank characters to erase parts of the old image at each move. If no blanking took place — using either method — then the effect would be spoiled by the image leaving a visible trail, the image is being drawn and redrawn to the same piece of electronic paper. The technique is summed up as draw, wait, erase and redraw. Block graphics are useful but a little boring, so fortunately we have access to pixel graphics.

```
10 REM BLOCK GRAPHIC ANIMATION
20 CLS
30 LET A$="▙▄▟"
40 LET B$="   "
50 FOR I=0 TO 21
60 PRINT AT I,12;A$
70 PAUSE 5
80 PRINT AT I,12;B$
90 NEXT I
100 GO TO 50
```

Figure 1

Pixels

Pixel graphics are much simpler for they are single points on the screen that can be turned on and off and, most importantly, their state may be tested. Pixel graphics introduce the idea of resolution — the higher the resolution (the number of pixels in a given area) the better the image. The Sinclair Spectrum has a very reasonable resolution of 176 by 256 pixels each of which can be individually set and tested. In case of a single pixel being moved around the screen the technique is similar to block graphics, the pixel is set and unset (erased) and moved to a new position after appropriate delay. Groups of pixels make up the high resolution screens that form the bulk of a computer games attractions. These shapes can be animated by dealing with the individual pixels that make them up. Collision detection is a computer inspired technique where the collision between two objects on the screen can be detected and the appropriate action taken. Shapes can be made to rebound from obstacles, a method used to great effect in bat and ball games. Collisions can lead to explosions which is again a widely used technique in the most popular arcade games.The program in figure two illustrates pixel animation and collision detection.

Some computers support Sprite graphics, unfortunately the Spectrum does not, although some users have managed to write interrupt driven software that gives a fair implementation of this feature which is normally supported by a dedicated chip. A Sprite is a shape that is defined and moved around the screen by dedicated software, and usually suports full collision detection. The program in figure two has simple

collision detection where the place the pixels will occupy next is teasted to see if it is possible to move there. A realistic rebound is programmed in rather than a straight reversal of course if a collision is detected.

Lines

Line graphics can be the most satisfying because they can be used to make up wire images of objects, giving a three dimensional effect. That object can be moved realistically around the screen and even rotated around variour points. Wire images have been used in some computer games but are probably best known in such applications as building and car design. The program in figure 3 is an example. These are the three main ways of producing graphics on most home computers and fortunately we have access to all three on the Sinclair Spectrum. It has to be said that the best animation on the Spectrum will be written using machine code rather than BASIC because machine code is a good deal faster. BASIC is easy to understand, and has been used in the examples here for clarity, though it can cause flickering and other odd effects and detract from the effect that you are trying to achieve

The actual process of animation on computers is a good deal different from that using hand drawn images and film. A computer really only undertands numbers, so that images shown have to be digitised to coordinates and these coordinates manipulated to produce new images in different positions. Do that quickly enough and you have animation. In extreme cases a single drawing may be made and many different images made by mathematical manipulation. For example, an image may be made wider or taller by altering values along one axis, give a computer two images, one bigger than the other and the computer can be programmed to produce a new image between the two sizes — this is known as an in-betweening. Animation implies change, so that images presented sequentially appear to move. The basis of animation involves a number of key elements that help produce the desired effect. The obvious is change in size or shape, others include change in speed, angle, colour, position and perspective — remember these are animation techniques and not computer techniques. We have all seen some really funny cartoons where some of the humour has been about the ability of a character or object to survive impossibly destructive situations — for example holding a bomb when it goes off, being squashed flat or streched. Twist, skew, shear, bend and explode can all be used to great effect. Explosions seem to charactersise the even popular arcade game and it is worth knowing how to do this.

Bang!

The basis of an explosion effect is that the image breaks up and moves apart from a central point. The image is initially chopped or minced into short lines or pieces and these move radially out from a central point. Most explosions are a variation of this, for example the bits of the exploded image can be made to tumble, flicker or leave trailing images as it moves apart. The image may be exploded into single pixels that change colour rapidly as they move outwards. A computer can generate useful sound effects to accompany this visual extravaganza which makes the whole process more effective.

Manipulation in size is one technique that a computer can do quickly without the need for new images to be drawn, digitised and stored. It is simply a matter of arithmetic to alter the size of an object. For example characters can be made larger by scanning the character square and reproducing the image larger or smaller. The programe is figured four does this four one character and displays an image twice the size at a different place on the screen. It achieves this by scanning all the sixty four pixels in the character square and displaying them doubled up on both axis. It is a simple matter to adapt this program to reproduce a character that is stretched taller or wider by only adjusting the values on one or the other axis. This technique, because it works at the pixel level will also expand any shape. In BASIC it is rather slow, in machine code it is much faster and can be used to great effect.

```
  10 REM PIXEL GRAPHIC ANIMATION
  20 REM AND COLLISION DETECTION
  30 CLS
  40 REM DRAW BORDER
  50 FOR I=1 TO 20: PRINT AT I,1
;"■": PRINT AT I,30;"■"
  60 NEXT I
  70 FOR I=1 TO 30: PRINT AT 1,I
;"■": PRINT AT 20,I;"■"
  80 NEXT I
  90 GO TO 220: REM MAIN PART
 100 REM PLOT SUBROUTINE
 110 PLOT X,Y
 120 PAUSE 4
 130 RETURN
 140 REM UNPLOT SUBROUTINE
 150 PLOT  OVER 1;X,Y
 160 RETURN
 170 REM COLLISION DETECTION
 180 REM SUBROUTINE
 190 IF POINT (X+XDIR,Y+YDIR)=1
AND POINT (X+XDIR,Y+YDIR+1)=1 AN
D POINT (X+XDIR,Y+YDIR-1)=1 THEN
 LET XDIR=XDIR*-1
 200 IF POINT (X+XDIR,Y+YDIR)=1
AND POINT (X+XDIR+1,Y+YDIR)=1 AN
D POINT (X+XDIR-1,Y+YDIR)=1 THEN
 LET YDIR=YDIR*-1
 210 RETURN
 220 REM MAIN LOOP
 230 LET X=70: LET Y=100: LET XD
IR=-3.1: LET YDIR=1
 240 GO SUB 170: REM COLLIDED?
 250 GO SUB 110: REM PLOT
 260 GO SUB 140: REM UNPLOT
 270 LET X=X+XDIR
 280 LET Y=Y+YDIR
 290 GO TO 240
```

Figure 2.

Perhaps the most exciting possibility for a home computer user is to make use of line graphics and produce a three-dimensional representation of an object that can be rotated in any given direction and create some illusion of movement. The wire frame image depends on coordinate geometry to manipulate it, it is a matter of numbers and few rules. The biggest problem is that the television or other computer screen is only two-dimensional. Perhaps one day hologram technology will have advanced enough so that three-dimensional images can be projected into a living room like a super television, but until then we must rely on the eye being fooled by two-dimensional trickery. Perspective drawings do this and the Renaissance artists were masters of the art and produced some remarkable paintings and drawings. The keys to such works of art is the 'vanishing point'. Looking along railway lines, the tracks appear to converge at a point — this is a vanishing point. These drawings are made with reference to such a point which is often off the paper.

3D

In three-dimensional reality we can describe an object by reference to three axis- x,y and z as they are conventionally known. When measuring an object we normally described the dimensions in terms of length, width and height, but in order to describe an object in terms suitable for computer graphics we must describe the object in distances from an origin point. The origin can be a corner of the object itself or further away, the object may be viewed from the origin or another point. This introduces the idea of how we can ▶

```
  10 REM CUBOID DRAWING
  20 LET OX=50: LET OY=40
  30 LET A=30: LET B=40: LET C=15: LET D=20
  40 PLOT OX,OY
  50 DRAW 0,A: DRAW B,0: DRAW 0,-A: DRAW -B,0
  60 DRAW C,D: DRAW B,0: DRAW -C,-D
  70 PLOT OX+0,OY+A: DRAW C,D: DRAW B,0: DRAW -C,-D
  80 PLOT OX+C,OY+D: DRAW 0,A: DRAW B,0: DRAW 0,-A
  90 LET A=A+1: LET B=B+1: LET C=C+1: LET D=D+1
 100 STOP
```

Figure 3

```
  10 REM DOUBLE SIZE CHARACTERS
  20 LET XC=1: LET YC=1
  30 PRINT AT 10,10;"A"
  40 DIM A(8,8)
  50 REM SCAN CHARACTER
  60 FOR X=0 TO 7
  70 FOR Y=0 TO 7
  80 IF POINT (X+80,Y+88)=1 THEN LET A(X+1,Y+1)=1
  90 NEXT Y
 100 NEXT X
 110 REM REPRODUCE ENLARGED
 120 FOR X=1 TO 16
 130 FOR Y=1 TO 16
 140 LET YC=INT (Y/2): IF YC=0 THEN LET YC=1
 150 IF A(XC,YC)=1 THEN  PLOT X+96,Y+84
 160 LET XC=INT (X/2): IF XC=0 THEN LET XC=1
 170 NEXT Y
 180 NEXT X
```

Figure 4

make a flat two-dimensional representation appear three-dimensional and provide the basis for animating such an object. The diagram in figure 5 shows how a single point viewed from the origin of the coordinates(x,y and z) can be represented on the picture plane at a set distance from the viewpoint. The picture plane of course is our television screen and the x and y coordinates of the point on it with respect to the bottom left hand coner are calculated as x times (distance of picture plane to viewpoint axis divided by z). This is called transforming and is good for any object where the viewpoint and the origin of the coordinate system are the same place.

Some degree of animation is now possible once the three-dimensions are transformed into two on the picture plane. The x,y coordinates can be transformed around the screen — enlarged, reduced, moved or rotated and the perspective altered by shifting the picture plane. Rotation presents an interesting problem because one has to resort to sines and cosines to solve the shifting of x and y coordinates about the centre of rotation. Taking an anticlockwise rotation through a *specified angle, A*, then the new coordinates of the point x,y will be calculated as x cosine *A*-y sine *A* for the x coordinate, and x sine *A* + y sine *A* for the y coordinate. This idea is incorporated

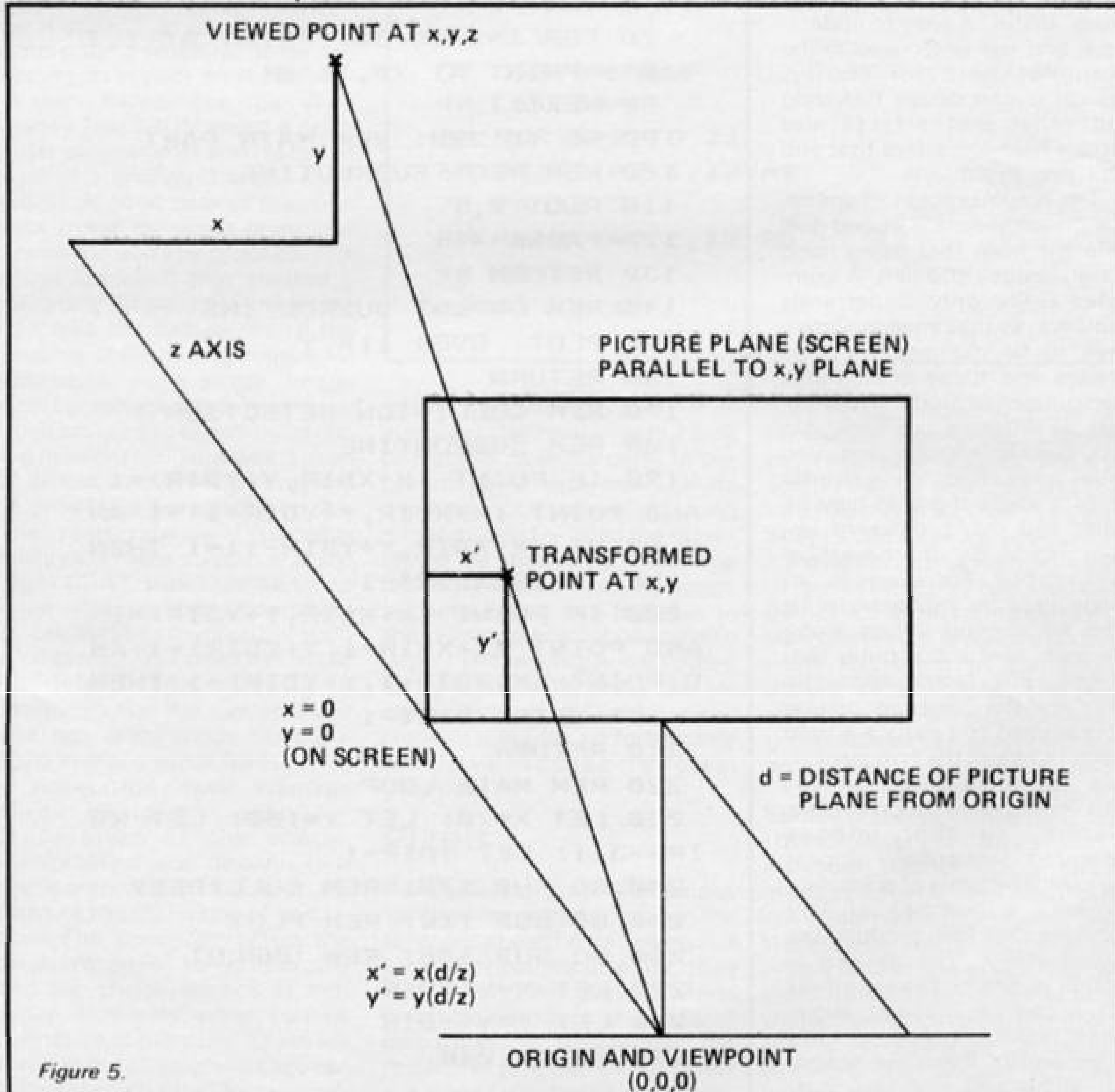
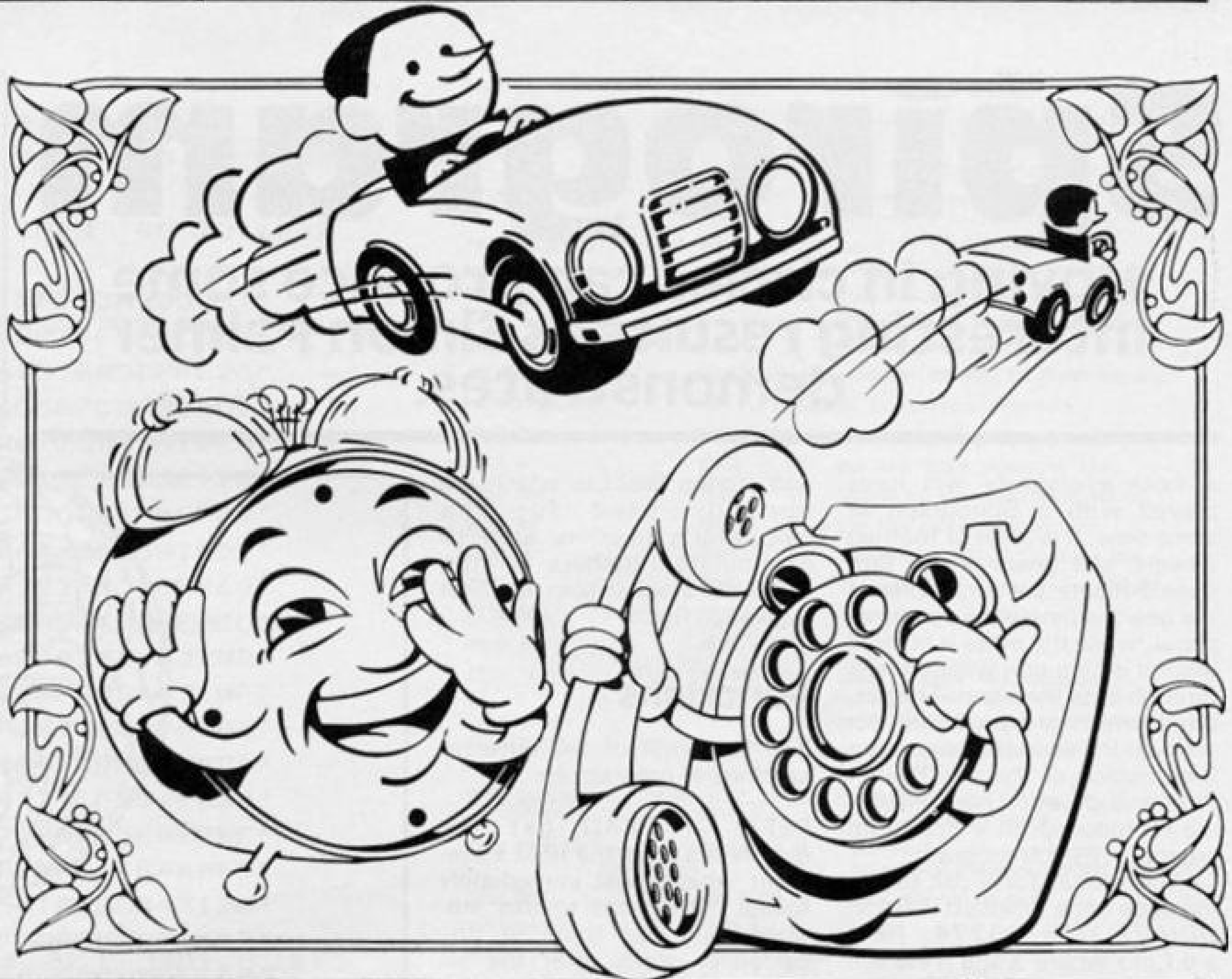


Figure 5.

into the program in figure 6 which draws a cube then rotates into a new position.

Line graphics are particularily suitable for these sorts of transformation, though it must be said that the mathamatics can become quite complex when trying to view the object from any point other than the origin of the coordinate system. Naturally for flicker free animation machine code is essential as there may be a large number of calculations involved in transforming each point on an object. Combinations of block, pixel and line graphics can be seen in many animated computer games as well as more serious applications but it must be said that really complex animation is the preserve of much larger and more expensive computers. What is needed in the home computer market is dedicated software that will allow easy and fast animation by coping with the transformations necessary, the software can be rather like the sprite graphics support available on a chip in some computers.

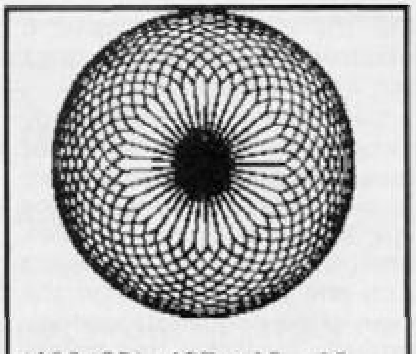


```
  10 REM SIMPLE ROTATION OF
  20 REM CUBOID SHAPE
  30 LET ANGLE=-.15: REM APPROX
18 DEGREES CLOCKWISE
  40 DIM X(8): DIM Y(8)
  50 REM COORDINATES
  60 DATA 80,80,80,110,120,110,1
20,80
  70 DATA 95,100,95,130,135,130,
135,100
  80 REM NOMINATE CENTRE OF
  90 REM ROTATION AT 100,110
 100 FOR I=1 TO 8
 110 READ X(I),Y(I)
 120 NEXT I
 130 GO SUB 190: REM PLOT SHAPE
 140 REM MAKE RELATIVE TO CENTRE
 150 FOR I=1 TO 8
 160 LET X(I)=X(I)-100: LET Y(I)
=Y(I)-110
 170 NEXT I
 180 GO TO 410: REM TO ROTATE
 190 REM DRAW CUBOID SUBROUTINE
 200 CLS
 210 PLOT X(1),Y(1)
 220 DRAW X(2)-X(1),Y(2)-Y(1)
 230 DRAW X(3)-X(2),Y(3)-Y(2)
 240 DRAW X(4)-X(3),Y(4)-Y(3)
 250 DRAW X(1)-X(4),Y(1)-Y(4)
 260 PLOT X(5),Y(5)
 270 DRAW X(6)-X(5),Y(6)-Y(5)
 280 DRAW X(7)-X(6),Y(7)-Y(6)
 290 DRAW X(8)-X(7),Y(8)-Y(7)
 300 DRAW X(5)-X(8),Y(5)-Y(8)
 310 PLOT X(4),Y(4)
 320 DRAW X(8)-X(4),Y(8)-Y(4)
 330 DRAW X(5)-X(8),Y(5)-Y(8)
 340 PLOT X(1),Y(1)
 350 DRAW X(5)-X(1),Y(5)-Y(1)
 360 PLOT X(2),Y(2)
 370 DRAW X(6)-X(2),Y(6)-Y(2)
 380 PLOT X(3),Y(3)
 390 DRAW X(7)-X(3),Y(7)-Y(3)
 400 RETURN
 410 REM APPLY ROTATION
 420 REM TO COORDINATES
 430 FOR I=1 TO 8
 440 LET X(I)=(X(I)*COS ANGLE)-(
Y(I)*SIN ANGLE)
 450 LET Y(I)=(X(I)*SIN ANGLE)+(
Y(I)*COS ANGLE)
 460 NEXT I
 470 REM CALCULATE SCREEN COORDS
 480 FOR I=1 TO 8: LET X(I)=X(I)
+100: LET Y(I)=Y(I)+110
 490 NEXT I
 500 GO SUB 190: REM PLOT AGAIN
 510 STOP
```

Figure 6.

Spirogram

Moving in circles can produce some interesting results as Simon Palmer demonstrates.



Almost everybody will have played with a Spirograph at some time. It is a set of toothed wheels, like gears which produce intricate patterns. One of the gears is pinned to a piece of paper, while the other is pushed around it by a pen which pokes through onto the paper. The patterns which are produced depend on the sizes the gears, and the position of the pen hole in the moving gear. These effects are reproduced on a Spectrum computer by this program.

The format for a call to the machine code program is RANDOMIZE USR 65274: REM a,b,f,m,r where a and b are the coordinates of the centre of the pattern on the screen, f and m are integers which are the radii of the fixed and moving gears, and r is the distance of the pen hole from the centre of the moving gear.

The program is based almost entirely on the Spectrum Basic interpreter's calculator (called by a RST 0028). This is controlled by a string of literals immediately following the call which describe what needs to be done, adding multiplying and so on. Particular use has been made of the number fetching and storing facilities which are normally used for the calculator's memory area to manipulate numbers on the calculator stack (a plan of which is given in figure 1).

Functions

The functions of the different sections of program are:

FEFA — FF40 get the arguments from the REM statement which must immediately follow a USR call to this program, and put them on the calculator stack after the six dummy values at the bottom to replace the calculator's memory area, of at any time an error is detected a jump is made to 1C8A which stops the BASIC program and produces an error report.

FF41 — FF54 calculates the hcf of f and m.

FF55 — FF74 calculates the size of angular increments △θ and the total number of them in a complete pattern.

FF75 — FF8A sets the initial values of COSθ, SINθ, COSØ, SINØ(COS & SIN denote (f + m) cosθ& (f + m) sinθ, and COSØ& SINØdenote r cosØand r sinØ) and calculates cos△θ, sin△θ, cos△Øand sin△Ø.

FF8B — FF9D increments θ (by changing COSθ and SINθ).

FF9E — FFAF increments Ø (by changing COSØ and SINØ).

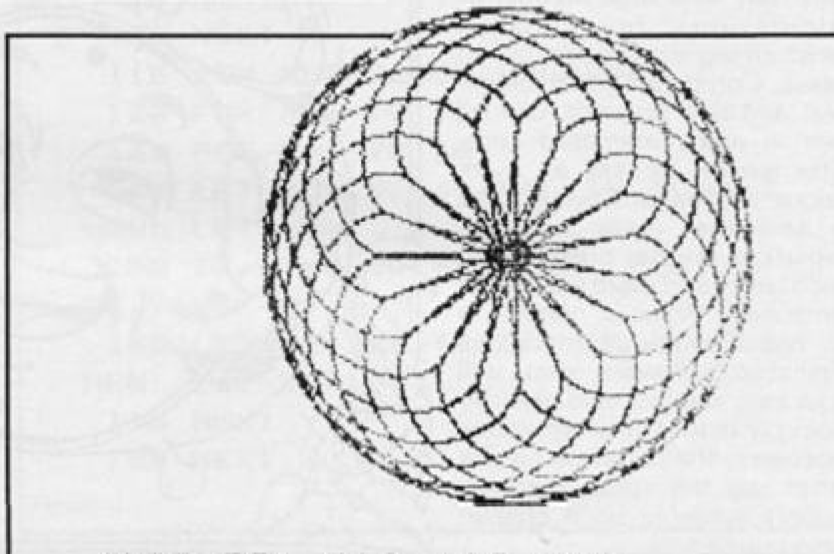
FFB0 — FFBB calculates the plot position (x,y).

FFBC — FFE2 checks to see whether (x,y) is on the screen, if so it is PLOTted by a USR call to 22DC.

FFE3 — FFF2 decrements the step count, if this is not zero and there is no keyboard break, then the loop is executed again.

FFF3 — FFFE resets the calculators memory area and clears the stack.

FFFF returns to BASIC.



Stack Map

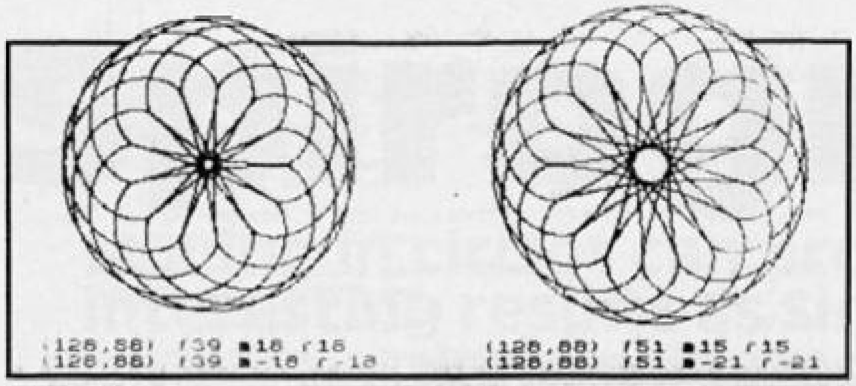
set-mem	get-mem	contents
C0	E0	calc0
C1	E1	calc1
C2	E2	calc2
C3	E3	calc3
C4	E4	calc4
C5	E5	calc5
C6	E6	a
C7	E7	b
C8	E8	f
C9	E9	m
CA	EA	r
CB	EB	t
CC	EC	△θ
CD	ED	COSθ
CE	EE	SIN θ
CF	EF	cos △θ
D0	F0	sin △θ
D1	F1	COSØ
D2	F2	SIN Ø
D3	F3	cos △Ø
D4	F4	sin △Ø
D5	F5	x
D6	F6	y

```
  1Ø REM SPIROGRAM
  2Ø CLEAR 65273
  3Ø LET P=1+PEEK 2373Ø+256*PEEK
 23731
  4Ø GO SUB 1ØØØ
  5Ø INPUT "(";a;",";b;") f";f;"
 m";m;" r";r
  6Ø RANDOMIZE USR P: REM a,b,f,
m,r
  7Ø GO TO 5Ø
1ØØØ REM poke SpiroGram
1Ø1Ø LET d=p
1Ø2Ø DEF FN d(d$)=(d$<="9")*(COD
E d$-CODE "Ø")+(d$>="A")*(1Ø+COD
E d$-CODE "A")
1Ø3Ø READ d$
1Ø4Ø POKE d,FN d(d$(1))*16+FN d(
d$(2))
```

```
1050 LET d=d+1
1060 LET d$=d$(3 TO )
1070 IF d$<>"" THEN  GO TO 1040
1080 READ d$
1090 IF d$<>"" THEN  GO TO 1040
1100 RETURN
1200 DATA     "DFFE3A2039E7"
1210 DATA "FEEA2034E5FDCB01"
1220 DATA "BE0605180 5DFFE2C"
1230 DATA "2026C5E7CDFB24C1"
1240 DATA "FDCB0176281A10ED"
1250 DATA "FDCB017E2015E122"
1260 DATA "5D5CFDCB01FEEFA0"
1270 DATA "A0A0A0A0A03818D1"
1280 DATA "C38A1C2A635C2268"
1290 DATA "5CEFE82AE92AEBEC"
1300 DATA "0336000201EB3202"
1310 DATA "313700F902A3310F"
1320 DATA "310F1B31E8E905A1"
1330 DATA "0FEA042AE8E90F2A"
1340 DATA "0F042731E904EB05"
1350 DATA "2A27CB0205E8E90F"
1360 DATA "A0EC20EC1FEAA0E8"
1370 DATA "E905A10FEC043120"
1380 DATA "011F38EFEEEF04ED"
1390 DATA "F0040FEDEF04EEF0"
1400 DATA "0403CD02CE02F2F3"
1410 DATA "04F1F4040FF1F304"
1420 DATA "F2F40403D102D202"
1430 DATA "E6ED0FF10F27E7EE"
1440 DATA "0FF20F27F5360022"
1450 DATA "F636001EF53440B0"
1460 DATA "00FF03370014F634"
1470 DATA "40B000AF0337000A"
1480 DATA "3480B000DC222D33"
1490 DATA "020202EBA103CB37"
1500 DATA "0004381807 38CD54"
1510 DATA "1F389821925C2268"
1520 DATA "5C2A635C22655CC9"
1530 DATA ""
```

```
FEFA DF        PROG  RST 0018        fetch char being interp'd
FEFB FE3A            CP A,3A         is it a ":"
FEFD 2039            JR NZ,ERROR     if not jump to ERROR
FEFF E7              RST 0020        fetch next char to be interp'd
FF00 FEEA            CP A,EA         is it a "REM"
FF02 2034            JR NZ,ERROR     if not jump to ERROR
FF04 E5              PUSH HL         the address of the arguments
FF05 FDCB01BE        RES 7,(IY+01)   reset bit 7 of (FLAGS)
                                     to check syntax of arguments
FF09 0605      ARGS  LD B,05         number of arguments
FF0B 1805            JR FIRST        jump to first, there is no
                                     separater before first argument
FF0D DF        NEXT  RST 0018        fetch char being interp'd
FF0E FE2C            CP A,2C         is it a ","
FF10 2026            JR NZ,ERROR     if not jump to ERROR
FF12 C5        FIRST PUSH BC         the argument count
FF13 E7              RST 0020        fetch next char to be interp'd
FF14 CDFB24          CALL 24FB       call the expression evaluator
```

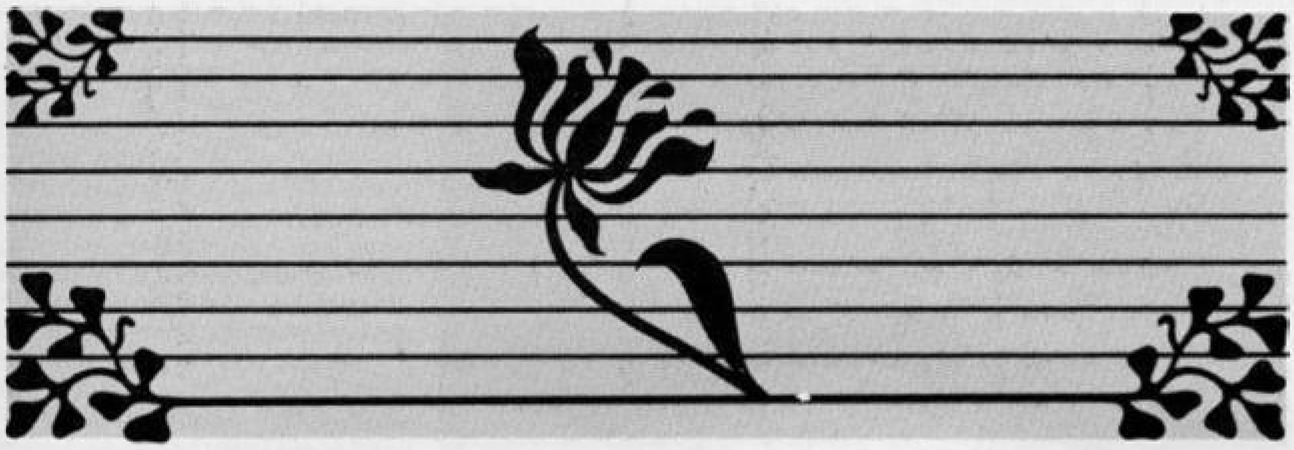
```
FF17 C1              POP BC          the argument count
FF18 FDCB0176        BIT 6,(IY+01)   set (NZ) if numeric expression
                                     reset (Z) if string expression
FF1C 281A            JR Z,ERROR      jump to ERROR if string exp.
FF1E 10EF            DJNZ NEXT       decrement argument count,
                                     if more to do jump to next
FF20 FDCB017E        BIT 7,(IY+01)   set (NZ) if interpreting
                                     reset (Z) if syntax checking
FF24 2015            JR NZ,BEGIN     jump to BEGIN if arguments have
                                     been stacked by expr. evaluator
FF26 E1              POP HL          address of arguments
FF27 225D5C          LD (5C5D),HL    restore (CH-ADD) to first argument
FF2A FDCB01FE        SET 7,(IY+01)   to interpret arguments
FF2E EF              RST 0028        call the calculator
FF2F A0              DEFB            put six dummy values at the
FF30 A0              DEFB            bottom of the calculator stack
FF31 A0              DEFB            for the calculator to use
FF32 A0              DEFB            instead of MEMBOT (enables the
FF33 A0              DEFB            set-mem C0,C1,.. & get-mem E0,E1,..
FF34 A0              DEFB            literals to be used)
FF35 38              DEFB end-calc
FF36 18D1            JR ARGS         jump to ARGS, add arguments
                                     a,b,f,m,r to the stack in
                                     positions 6,7,8,9,A respectively
FF38 C38A1C    ERROR JP 1C8A         stop program and produce
                                     error report
FF3B 2A635C    BEGIN LD HL,(STKBOT)  address of the bottom of
                                     the calculator stack
FF3E 22685C          LD (MEM),HL     address of the area used for
                                     calculators memory (usually MEMBOT)
FF41 EF              RST 0028        call the calculator
FF42 E8              DEFB get f
FF43 2A                   absolute  |f|
FF44 E9                   get m
FF45 2A                   absolute  |m|
FF46 EB                   get |f|
FF47 EC                   get |m|
FF48 03                   subtract  |f|-|m|
FF49 36                   less-0
FF4A 0002                 jump if |f|<|m| to ORDRD
FF4C 01        UCLID      exchange  so that larger number l is on
                                    top of smaller number s
FF4D EB        ORDRD      get s
FF4E 32                   l-mod-s   returns remainder and quotient
                                    on top of stack
FF4F 02                   delete    the quotient
FF50 31                   duplicate the remainder
FF51 37                   greater-0
FF52 00F9                 jump if remainder. 0 to UCLID
FF54 02                   delete    the zero remainder
FF55 A3                  const-π/2
FF56 31                  duplicate
FF57 0F                  add       π
FF58 31                  duplicate
FF59 0F                  add       2π
FF5A 1B                  negate   -2π
FF5B 31                  duplicate
FF5C E8                  get f
FF5D E9                  get m
FF5E 05                  divide    f/m
FF5F A1                  const-1
FF60 0F                  add       1+ f/m
FF61 EA                  get r
FF62 04                  multiply  r(1+ f/m)
FF63 2A                  absolute |r(1+ f/m)|
FF64 E8                  get f
FF65 E9                  get m
FF66 0F                  add       f+m
FF67 2A                  absolute |f+m|
FF68 0F                  add      |f+m| + |r(1+ f/m)|
FF69 04                  multiply      -2π(|f+m| + |r(1+ f/m)|)
FF6A 27                  integer  INT( -2π(|f+m| + |r(1+ f/m)|))
```




```
FF6B 31                    duplicate number of steps in one circuit
FF6C E9                    get m
FF6D 04                    multiply
FF6E EB                    get hcf
FF6F 05                    divide
FF70 2A                    absolute
FF71 27                    integer   m/hcf INT(-2π(|f+m| + |r(1+ f/m)|))
FF72 CB                    set t     number of steps in complete pattern
FF73 02                    delete
FF74 05                    divide    -2π/INT( -2π(|f+m| + |r(1+ f/m)|))
                                     largest angular increment which will
                                     not leave gaps in the pattern
FF75 E8                    get f
FF76 E9                    get m
FF77 0F                    add       f+m → COS θ → mem-0D
FF78 A0                    const-0   0   → SIN θ → mem-0E
FF79 EC                    get Δθ
FF7A 20                    cos       cos Δθ      → mem-0F
FF7B EC                    get Δθ
FF7C 1F                    sin       sin Δθ      → mem-10
FF7D EA                    get r     r   → COS Ø → mem-11
FF7E A0                    const-0   0   → SIN Ø → mem-12
FF7F E8                    get f
FF80 E9                    get m
FF81 05                    divide    f/m
FF82 A1                    const-1
FF83 0F                    add       1+ f/m
FF84 EC                    get Δθ
FF85 04                    multiply  ΔØ = (1+ f/m)Δθ
FF86 31                    duplicate
FF87 20                    cos
FF88 01                    exchange  cos ΔØ      → mem-13
FF89 1F                    sin       sin ΔØ      → mem-14
FF8A 38          DEFB      end-calc
FF8B EF     LOOP RST 0028  call the calculator
FF8C EE          DEFB get SIN θ
FF8D EF               get cos Δθ
FF8E 04               multiply  SIN θ cos Δθ
FF8F ED               get COS θ
FF90 F0               get sin Δθ
FF91 04               multiply  COS θ sin Δθ
FF92 0F               add       SIN(θ+Δθ) = SIN θ cos Δθ + COS θ sin Δθ
FF93 ED               get COS θ
FF94 EF               get cos Δθ
FF95 04               multiply  COS θ cos Δθ
FF96 EE               get SIN θ
FF97 F0               get sin Δθ
FF98 04               multiply  SIN θ sin Δθ
FF99 03               subtract  COS(θ+Δθ) = COS θ cos Δθ - SIN θ sin Δθ
FF9A CD               set COS θ
FF9B 02               delete
FF9C CE               set SIN θ
FF9D 02               delete
FF9E F2               get SIN Ø
FF9F F3               get cos ΔØ
FFA0 04               multiply  SIN Ø cos ΔØ
FFA1 F1               get COS Ø
FFA2 F4               get sin ΔØ
FFA3 04               multiply  COS Ø sin ΔØ
FFA4 0F               add       SIN(Ø+ΔØ) = SIN Ø cos ΔØ + COS Ø sin ΔØ
FFA5 F1               get COS Ø
FFA6 F3               get cos ΔØ
FFA7 04               multiply  COS Ø cos ΔØ
FFA8 F2               get SIN Ø
FFA9 F4               get sin ΔØ
FFAA 04               multiply  SIN Ø sin ΔØ
FFAB 03               subtract  COS(Ø+ΔØ) = COS Ø cos ΔØ - SIN Ø sin ΔØ
FFAC D1               set COS Ø
FFAD 02               delete
FFAE D2               set SIN Ø
FFAF 02               delete
FFB0 E6               get a
FFB1 ED               get COS θ
FFB2 0F               add
FFB3 F1               get COS Ø
FFB4 0F               add
FFB5 27               integer   x = INT(a+ COS θ + COS Ø)
                                  = INT(a+ (f+m) cos θ + r cos Ø)
                                    mem-15
FFB6 E7               get b
FFB7 EE               get SIN θ
FFB8 0F               add
FFB9 F2               get SIN Ø
FFBA 0F               add
FFBB 27               integer   y = INT(b+ SIN θ + SIN Ø)
                                  = INT(b+ (f+m) sin θ + r sin Ø)
                                    mem-16
FFBC F5               get x
FFBD 36               less-0
FFBE 0022             jump to NPLOT if x<0
FFC0 F6               get y
FFC1 36               less-0
FFC2 001E             jump to NPLOT if y<0
FFC4 F5               get x
FFC5 3440B000FF       stk-data 255 decimal
FFCA 03               subtract
FFCB 37               greater-0
FFCC 0014             jump to NPLOT if x>255dec
FFCE F6               get y
FFCF 3440B000AF       stk-data 175 decimal
FFD4 03               subtract
FFD5 37               greater-0
FFD6 000A             jump to NPLOT if y>175
FFD8 3480B000DC22     stk-data 22DC hexadecimal
                               address of PLOT x,y routine
FFDE 2D               USR call to PLOT routine
FFDF 3302             jump tp PLOTD
FFE1 02        NPLOT  delete y
FFE2 02        PLOTD  delete x or value returned by USR
                               if PLOT routine was executed
FFE3 EB               get t
FFE4 A1               const-1
FFE5 03               subtract
FFE6 CB               set t
FFE7 37               greater-0
FFE8 0004             jump to CONT if t>0
FFEA 38               end-calc
FFEB 1807        JR END
FFED 38        CONT   end-calc
FFEE CD541F      CALL 1F54   return C (carry flag set)
                             unless 'break' i.e. caps shift
                             and space keys being pressed
FFF1 3898        JR C,LOOP   continue if no break
FFF3 21925C    END LD HL,5C92  =MEMBOT the address of the
                             calculators normal memory
FFF6 22685C      LD (MEM),HL
FFF9 2A635C      LD HL,(STKBOT) address of stack bottom
FFFC 22655C      LD (STKTOP),HL delete the stack by overwriting
                               the address of the stack top
FFFF C9          RET         return to BASIC
```

The UDG Bank

84 UDGs must be enough for anyone. B. J. Kamphuis shows us how to achieve this on a Spectrum.



User Defined Graphics are a valuable and relatively easy means of producing arcade action, background pictures or business graphic details. Although twenty one are easily obtainable, it is surprising how often you find you could do with a few more.

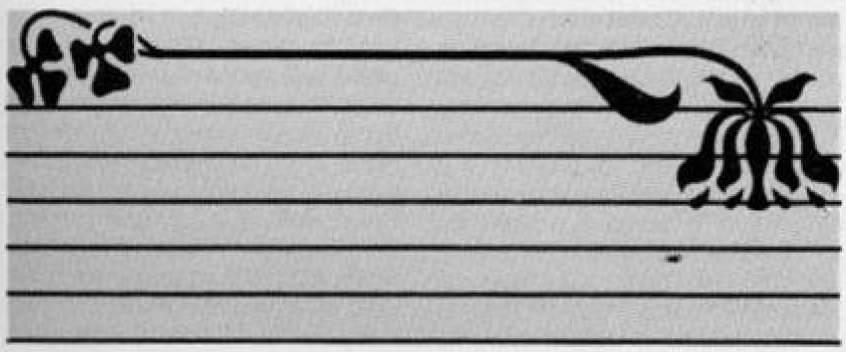
There are many ways of getting an increased number of UDGs and B. J. has adopted the method of storing four banks which are transferred to and from the set UDG area of memory as required. Now if 168 bytes have to be moved by BASIC a noticable delay is introduced so a short machine code routine is used to keep the program moving smoothly. The UDGs are stored in four banks named, not surprisingly, bank1, bank2, bank3 and bank4. Every bank can be transferred to the 'active' or normal UDG set and back.

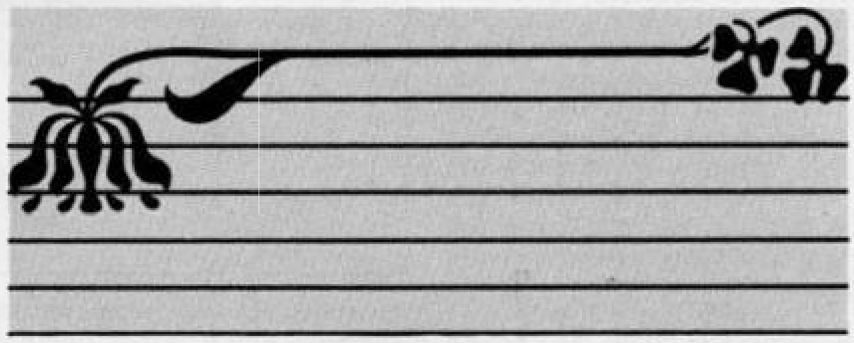
To define a bank all you need to do is set them up as ordinary UDGs from BASIC as normal, then use this program to store the characters in the memory reserved for that bank. To move a bank of UDGs to the storage area all you have to do is use the command, either from a BASIC program or as a direct command: **RANDOMIZE USR address: REM W1**

Now 'address' is either 64000 or 31232 depending on whether you have a 16K or 48K Spectrum, and the 'W' after the REM tells the machine code routine to write from the normal UDG area to store, and the '1' tells it which bank to use. Therefore when setting up your UDG banks you will use W1, W2, W3 and W4 depending on which bank you are working on.

The opposite effect, that of loading the normal UDG area with the characters in a bank is achieved by using **RANDOMIZE USR address: REM L1** — note 'L' instead of 'W'. Once the banks have been set up then save the whole four sets plus the machine code with the command SAVE "udgs" CODE address, 1535 and to use in a program include after the program has loaded the commands CLEAR address - 1: LOAD""CODE.

```
   1 REM
   2 REM 16K version
   3 REM
  10 CLEAR 31231: DATA 42,93,92,
35,35,126,254,87,202,30,122,254,
76,202,60,122,201,35,126,214,48,
33,88,127,17,0,122,130,87,1,168,
0,237,176,201,35,126,214,48,17,8
8,127,33,0,122,132,103,1,168,0,2
37,176,201
  20 FOR f=31232 TO 31248: READ
a: POKE f,a: NEXT f
  30 FOR f=31262 TO 31279: READ
a: POKE f,a: NEXT f
  40 FOR f=31292 TO 31309: READ
a: POKE f,a: NEXT f
```

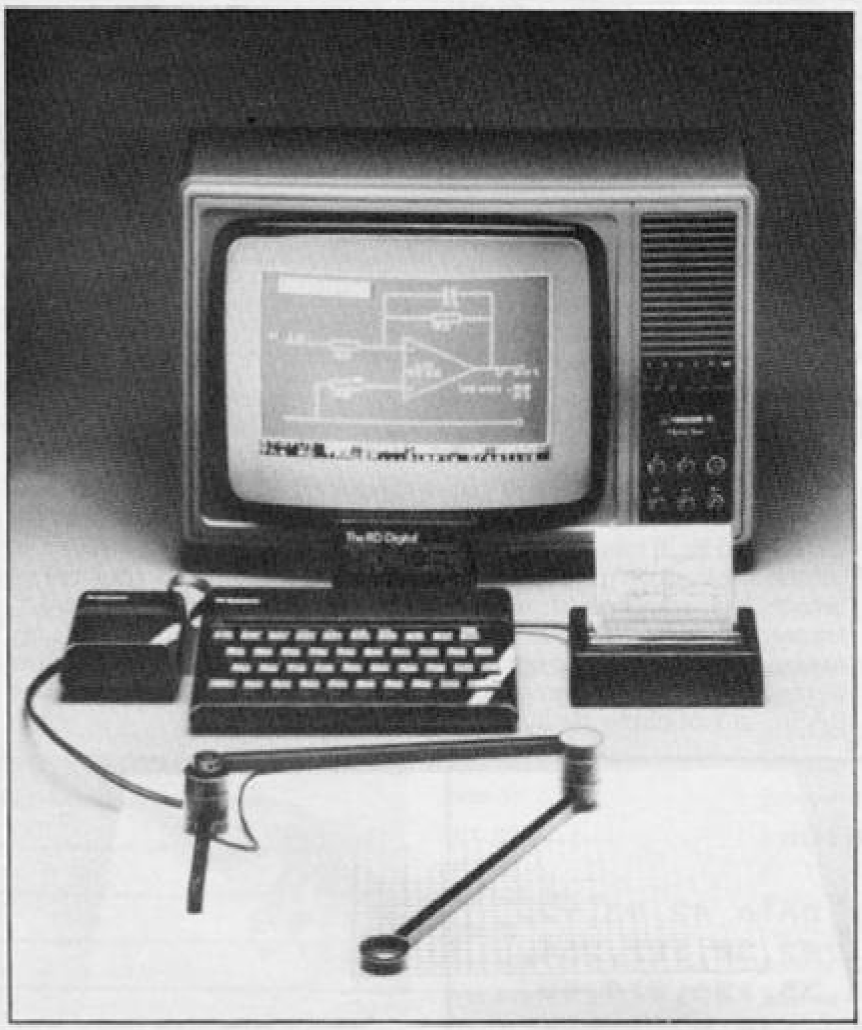




```
   1 REM
   2 REM 48K version
   3 REM
  10 CLEAR 63999: DATA 42,93,92,
35,35,126,254,87,202,30,250,254,
76,202,60,250,201,35,126,214,48,
33,88,255,17,0,250,130,87,1,168,
0,237,176,201,35,126,214,48,17,8
8,255,33,0,250,132,103,1,168,0,2
37,176,201
  20 FOR f=64000 TO 64016: READ
a: POKE f,a: NEXT f
  30 FOR f=64030 TO 64047: READ
a: POKE f,a: NEXT f
  40 FOR f=64060 TO 64077: READ
a: POKE f,a: NEXT f
```

Graphics '85

A comprehensive review of the state of the art by Colin Christmas.



As we see this old year out and welcome the new one in, it's a better time than most to stand back and take stock. Looking back (and leaving the prophecies and predictions to others) it must surely be agreed that for Spectrum users with an eye on Graphics it has been a most exciting year. I can't speak for other departments but for me, it has been Christmas every issue.

In this issue I want to do something special — take a look at the goodies that have come our way. Our way? Who are We? We are the Graphics Grabbers, Artwork Addicts, Design Doodlers. For us the screen is a window on a world of colour, images, line, shape, pattern, design, texture, light and shade. And like a window it opens out into an exciting new visual world. For business, for profit, for learning for discovering or for pleasure. From the weekend dabbler or doodler to the serious professional, from the games enthusiast to the educational user, from beginner to experienced programmer and right across the age range, you will find Spectrum owners who are hooked on graphics.

It's fairly formidable task — covering the range of Graphics Hardware and Software now available to Spectrum owners but let's get started.

Software

PRINT N'PLOTTER Products have given us two thoroughbreds in graphics packages, stablemates in fact. Both are already household names in the world of Spectrum Graphics. *Paintbox,* redesigned and renamed *Paintplus* since first reviewed, and *Screen Machine.*

Paintplus now has over twenty new features added to the original. It is Microdrive and Grafpad compatible, and offers cursor control from either keyboard or joystick, swift and smooth or slow and smooth. You have a choice here too. Power to program up to eighty four user definable characters in the UDG Editor, experimenting all the time with the Sketch Pad facility before deciding to use them in screen graphics or programs. Also on the Menu, Precision Plotter enabling high resolution screen graphics work to be attempted and then combined with your already designed UDGs and stored in four UDG banks.

Multiple screen files can be achieved by working your way through the twenty eight page manual which accompanies the software cassette. The new features are found here in Precision Plotter. . . *Screen Planner* and *Organiser.* They include a mode called Paperwash, a facility for Grabbing from Screens, Box Draw, Enlarge, Edit modes and many more. One side of the cassette has demonstration programs to inspire the fainthearted.

This toolkit needs patient and diligent effort put in to it before those efforts are really rewarded. It is not for those who want an 'instant meal' to be taken on the move so to speak. It's not for doodlers either. Members of either group will enjoy the heart of the product only. To get the most out of it you need some application skills.

If *Paintplus* were an 'O' Level, then *Screen Machine* would be an 'A' Level. By that I mean that success with the one will probably lead to moving up to the other. It is capable, of lifting earlier work done say on Paintplus, to new and more professional heights.

The demo programs on the cassette are not only inspiring, as was the case with *Paintplus,* but in the case of *Screen Machine* are also meant to be instructive. The program for the example of Flip Screen Animation can, like the others, be broken into, so that you can see, or begin to see how this impressive feat was achieved. The manual explains too, how it was done, and with this and its other detailed explanations, instructions and suggestions, gives a clear indication right from the start of how powerful a utility *Screen Machine* really is.

It should be said that a good working knowledge of BASIC and of the Spectrum Manual itself is essential before trying to get to grips with this tool. One of its most innovative and useful facilities is that of enabling programmers to save time and memory space by having access to instant Machine Code routines for Compressing Screen displays, saved normally as 6912 Bytes, but now being stored as only 2475 bytes. There are sections too, in the manual, with instructions for adapting *Screen Machine* to microdrive, for a wide range of screen modifications and for text compilation into machine code.

These two from Print N'Plotter, then, a invaluable twosome, especially when teamed up together. They will not suit every style or requirement, so read on. . . .

From NEW GENERATION SOFTWARE came *Lightmagic.* And that might just suit some. It is easier to use, or rather to get straight into and, in the true sense of the word it is fun. If all you want to do to start off with is to make pictures, play with UDGs and so on, in a light hearted way that will be entertaining and will not tax your concentration too greatly at first, then *Lightmagic* is the friendly tool for the job. It is therefore a good starter pack in my opinion and I have observed its magnetic attraction in action with children aged eight to sixteen with only keyboard familiarity on the Spectrum.

The manual gets full marks from me. Clear, easy to read, well planned and with information, instructions, index, tips

and summaries all included in such a way as to show that a great deal of thought went in to its organisation and production. Text Mode and UDG Mode are fascinating and easy to work with. Text and characters appear at the bottom of the screen and can then be 'picked up' and positioned anywhere you want them in your picture or design.

Then, just when you will probably be ready to move on to more ambitious projects, graphically, Lightmagic proves its magic again. The program also offers a facility called UDG Designer for the creation and editing of your own graphics characters which you can then use in your own pictures and programs. The facility for Compressing pictures and saving memory is offered in the COMP-SCREEN option. This is loaded separately from the cassette. It's there in other words, when you need it. Later rather than sooner probably.

Alongside Lightmagic, not for comparison, but because for me they have the same feel about them, might come *Leonard*, from CREATIVE SPARKS. Again, full marks for the manual. Manuals ought to be user friendly as the software I think and in the case of *Leonardo* both are in fact true. I don't hesitate to recommend it as an excellent starter pack which is then capable of extending your own abilities as you learn to use it. A preliminary review of *Leonardo* appeared in the last issue of ZXC.

This is a good place to suggest that anyone who is into graphics or who wants to get started should get out and about into the shops to make their own evaluations. Failing that use the information in this article to put pen to paper and write off and ask for details of packages from the manufacturers.

Lightpens

I've had most success to date, with the package from Dk'tronics. The pen itself is rather like a biro or felt tip pen. It is attached by a wire to a control interface which of course comes with the package. The interface is plugged into the back of the Spectrum. A program on cassette is included.

The glass screen of your monitor is the working area and drawing surface, so some consideration has to be given as to whether this is the way you want to work. Then there are practical aspects such as the distance of your screen from your keyboard, and the fact that you have to work on a perpendicular 'face'. The height of the screen is therefore important if you do not want to suffer from muscle fatigue in your drawing arm.

Lightpens give you a physical contact and interaction with your drawing surface if that is important. Calibrating the pen each time may prove a chore, but after that it's plainsailing — within the limitations of the power of the program. Again it's a good way of getting started on the very basics of graphics, of getting into the picture as it were. Sensibly introduced in the classroom it could be useful aid and introduction for children in an educational context. It is limited though in its potential for advanced or complex screen designs. Graphics Tablets give you similar physical point of contact with your drawing surface. This time it is horizontal and again a 'pen' is used. There's a review of the *Saga* Graphics Pad in this issue. So when you are ready look it up. They certainly take you further than the lightpen. But then you pay a lot more for the facilities they offer.

Now for something almost completely different, the *Sinclair LOGO* pack. Another excellent starter, but as I have hinted, quite different.

This pack has very obvious educational applications and for very young children. The founding father of the LOGO language intended it as a language for children which would develop logical thinking, introduce young minds to computer programming and have very definite terms of reference for the teaching and development of mathematical concepts. Drawing is achieved by moving a small graphics 'turtle' — a triangle — around the screen. This is done by sending through the computer commands known as Primitive Procedures (mostly single words and abbreviations of those words). Your sense of direction needs to be accurate and formulated mathematically. Once you have established procedures for drawing, say, a square, this group of procedures can be assigned a single word or name which LOGO will then understand as a command to repeat the whole set of procedures.

The emphasis or bias is fundamentally mathematical, arithmetical or geometric. You do not just learn to draw a square, you also learn what makes a square what it is and from there the difference between a square and a rectangle or a parallelogram.

It is a language itself, apart from BASIC. Hence learning to use it is learning to program a computer in another language. The graphic aspect being displayed on the screen is part of the incentive and motivation for progressing with the new language.

Two fairly weighty and comprehensive books or manuals are part of the pack. The first book deals exclusively with Turtle Graphics and is an absorbing and refreshingly different kind of programming experience. The second book acts as a reference manual for *Sinclair LOGO*. The growth, use and development of LOGO by Spectrum owners, especially in schools will, I think, be affected by the cost factor.

When DREAM SOFTWARE released *Computer Aided Designer*, my own children had not had their Spectrum for long. They, like me were exploring the full graphics potential of the machine when *C. A.D.* turned up and kept us enthralled for days. Now, still an old favourite, I would recommend it as another in the 'Starter' category. With very obvious educational values and as a springboard for more ambitious projects later in Design.

The manual is simple and very straightforward — alphabetically leading you through the twenty seven commands available in the program. Some forty custom shaped graphics, UDGs can be designed. By giving precise measurements most geometric shapes can be drawn, filled and so on. It remains impressive after all this time, and the potential for drawing in 3D is considerable.

Similarly, another old favourite, *VU-3D* from PSION. This has the added and appeal of enabling the viewer to move around the object in 3D. Graphics and Design, pure and simple. High resolution colour and an incredible understanding of perspectives are real bonuses with this program.

Future designers in the Aircraft or for that matter almost any other Industry, may have started young with something like *C.A.D.* or *VU-3D*.

I doubt if they would have been able to afford the *RD Digital Tracer*, from RD Laboratories. This is closer to an instrument than anything else I've come across in graphics and design hardware and software for the Spectrum.

It comes in two versions, the Standard and the Professional. Both are fairly highly technical and sophisticated tools. The Tracer consists of a short fixed arm and pivot from which extends a drawing arm hinged at the centre with another floating pivot which moves across your drawing surface area.

The arm is connected to the computer by a length of cable via an interface plugged into the rear port of the Spectrum. A cardboard template and transparent grid overlay are included for calibration purposes. the tracer is a precision instrument. The software cassette contains five programs. The usual options are offered in the first, plotting single points, construction of basic geometric figures, filling, hatching, change of ink, border, paper colour, adding text, UDGs and so on.

The display image can be moved up, down, and from side to side, scaled up and down, and reversed. Multiple screen images including images at different scales and at different positions can be achieved. By adding other BASIC routines and software, the Tracer's capabilities can be extended into the field of statistical analysis. This immediately puts the Tracer into a specialist Graphics and Display category. Although the Tracer can be used with the ZX81 and 16K Spectrum, its full potential can only really be developed on the 48K and then only by competent programmers. It's a versatile instrument for the specialist.

It's the season of Good will and all that, so why not give a last mention for all whose speciality is Games Designing. It's been around for a while, but standing the test of time in lots of ways. I'm referring of course to the High level User Friendly Real time Games Designer from Melbourne House. Or as it is more commonly known, *HURG*.

Still a powerful program and a very good manual. How did they do it in those all-time greats like Pacman, Donkey Kong and Space Invaders? H.U.R.G. will tell you how.

It's a pretty good list of graphics goodies and that other seasonal expression comes to mind. 'There's something here for everyone.' You have no excuse for not knowing how and from whom in Spectrum Graphics. Just how to enjoy the graphics power behind those buttons. ■

Zapping Your Graphics

ZX looks at some utilities that can help put some life into your graphics displays.



White Lightning

This program has been around for some time now and I well remember the hours of studying the little, small typed, 131 page manual trying to get to grips with it. It's not that the manual is badly written, just the opposite, it's just that you have to throw away all your knowledge of programming in BASIC and learn what amounts to two complementary new languages.

Spectra Forth is a variant of the Forth language and its implementation is explained in detail, but there's no attempt to teach you how to program in Forth. This means that before you can get full benefit of this program you have to buy a book, read it and become competent in Forth.

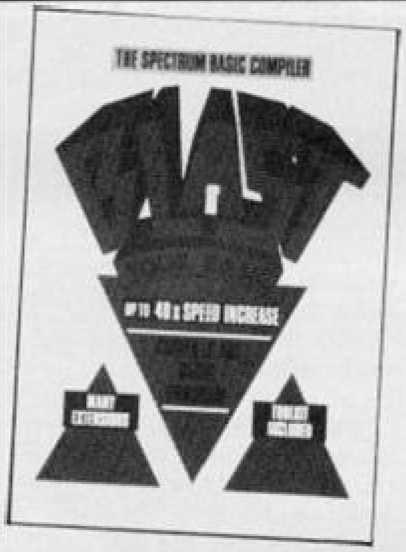
White Lightning not only gives you the use of a faster language but also over a hundred special commands in 'Ideal', a specially designed language created by Oasis.

The presentation case holds the previously mentioned manual plus two tapes containing White Lightning, a Sprite Development package and two demonstration programs. Look at the demos first, they will convince you that it may be well worth taking the time to learn to use the program.

White Lightning is a superb package for the dedicated games writer, it's good to see Forth having a practical use, and the end results could be superior to any other games designer programs.

Spectrum Laser Basic

Marketed by Ocean but written by Oasis, this is a very comprehensive package intended to 'enhance the ease, and particularly the speed, with which complex animated graphics can be produced'. Over 100 extra commands and functions are provided to this end and all operate from within BASIC. The program has a lot in common



with Oasis' *White Lightning*, but concentrating purely on graphics means that you do not have to learn Spectra Forth, as essentially the same commands are used as with their Ideal language.

There are a few changes, mostly names of commands to aid clarity, but basically this appears to be a stripped down, but still very powerful, version of *White Lightning*.

If you already have *White Lightning* then you will not get a great deal from this program, but, if you do not have *WL* and want a slighly simpler program which is essentially for graphics work then LASER BASIC is ideal. But be prepared to spend time learning to use it.

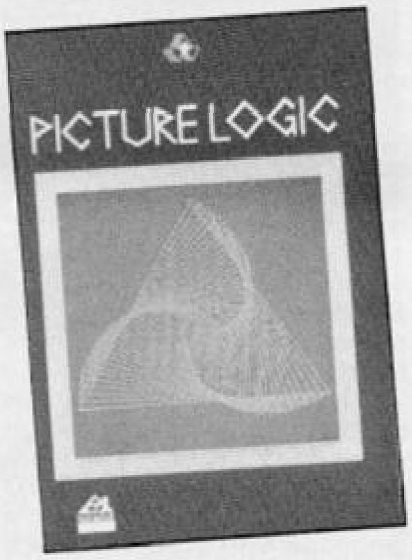
Ocean IQ, £14.95.

Picture Logic

This is an attempt to introduce LOGO to children and adults. As well as the program tape it contains two booklets, one aimed at adults and the other aimed at children — probably aged from around 12. As an introduction I feel it succeeds perfectly on both the levels attempted, the children's book has step by step instructions and exercises followed by challenges. It is well set out with screen pics and diagrams. The adults' book is written in less detail but is easily understood. Picture Logic supports 25 commands, variables and four arithmetical functions (addition, subtraction, multiplication and division).

This is a gentle and pleasantly presented introduction to LOGO, though I feel it is a little limited and you could soon reach the full extent of its capabilities. It would serve as an excellent introduction for schools, though it is a bit expensive.

Addison Wesley Publishers Ltd. £22.75



Blast

Although this is not technically or specifically a graphics program, I feel that much of its use will be for speeding up games and so decided to include it here. *BLAST* has been advertised for some time now and several versions exist.

The advertising seems to be a little optimistic, I was moved to comment in the last issue in the 'All Sinclair Machines' claim and of course it does not work on the QL or ZX81 but on all 48K versions of the Spectrum. It was an ambitious project and very nearly succeeds in their claim to 'compile any BASIC program', the program to do this takes up all but 2K of the Spectrum's memory. So now you have another problem, who wants to compile programs of only 2K or less? The answer that OCS came up with was to provide an extra TOOLKIT program. As a toolkit it is probably the most feeble available, however it allows you to save the program that you want to compile in a special format onto tape or microdrive.

So, you load Toolkit, load your program, save your program in the special form on tape then move on to phase two. This consists of loading BLAST and starting the compiler. This is very easy if time consuming, the program prompts for starting the tape and stopping, compiles each block, and when it is ready tells you to change tapes and save what it has done so far. This sounds complicated but is easy in practice. It is much easier on a twin microdrive system and OCS 'strongly urge' you to get one.

There are two forms of compiled code, p-code which is a sort of half way between machine code and BASIC and has the advantage of using less memory than either m/c or the original BASIC but, though faster than BASIC, is slower than m/c. The second is real machine code, this is usually larger than its Basic original (unusual) but faster than p-code.

A useful feature is AUTORUN which makes compiled programs hard to break and copy. Not so good is the fact that the planned extensions to BASIC had to be abandoned due to lack of room, to make up for it a few more mediocre additions were made to the toolkit program.

Despite some reservations, BLAST is by far the best compiler we have seen (so far), it worked perfectly with all the programs we tried. I'm not too sure I like the over-optimistic advertising claims and one or two of our readers have written to tell us of problems they have had. These do seem to be with specialist applications and if you are not intending to use it for general programs but for some with vast amounts of data for example, then it is best to phone or write to get confirmation that will do exactly what you require.

Oxford Computer Systems (Software) Ltd.
£24.95.

Doing It With Style



There are quite a few graphics utilities on the market at the moment, but one that particularly caught our attention (mainly because it came in the biggest box) was *Style* from Saga Systems.

Style costs £29.95, and though that might seem a bit steep, you do get a Kempston compatible joystick included in the price. The interface is required because the onscreen cursor that you draw with is controlled by either a joystick, or Saga's own Graphics Pad which can be bought separately for an additional £79.95. That might seem expensive, but it is still the cheapest pad around.

The pad is a ceramic tablet, rather like the old 'Etch-a-Sketch' toys. You can 'draw' onto this pad using the 'pressure pen' that is built into it, and your hand movements are translated directly onto the screen. This is much easier to use for freehand drawing than the clusters of keys that most graphics utilities use, though you do lose some of the pixel-fine precision of keyboard control as your hands are unlikely to be absolutely steady. However, *Style* does have a Zoom facility that allows you to magnify sections of the screen and tidy up any stray pixels caused by handshake.

To be perfectly honest, *Style* isn't the most versatile graphics utility that we've seen. The functions that it offers are all fairly standard; Box, Circle, Fill, and a few others. These are all quite well executed, especially the Box and Triangle options which allow you to move shapes around the screen, and to try out different sizes of shapes before you 'drop' them into place.

Where *Style* really scores, in conjunction with the Graphics Pad, is in ease of use. There are no massive manuals to plod through, and the icon driven menu is very simple to use. I find that most graphics packages require you to master some fairly finger tangling combinations of keys to move from drawing mode to another. With *Style*, all you have to do is press one button to return to the main menu, then position the cursor over the relevant icon (a small circle for 'Circle', a squiggle for 'Freehand' etc.) and the program automatically returns you to your picture and is ready to carry out the next command.

The instruction leaflet says that you can read it while Style is loading, and will probably not need to use it again. And, surprisingly, that proved to be the case. After just a few minutes trying out the icon menu I was happily doodling away on the Graphics Pad, drawing squiggly lines and boxes all over the place (it doesn't take much to keep me happy).

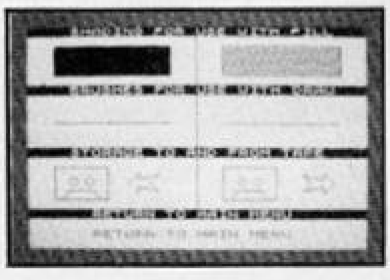
Actually, the ability to draw squiggly lines using the Pad is one area where *Style* does score over its competitors. I've always found trying to draw smooth curves on the screen, just using the clusters of keys that most utilities use for cursor control, very difficult. Keyboard control does allow you more precision, but slows you down a lot. The Graphics Pad loses some of that precision (though, as I've mentioned, the Zoom facility does allow you to tidy things up), but allows you to draw much more freely, as if you were drawing onto a sketch pad. For artistic failures like myself, who find keyboard control too fiddly, this is a big plus (though at a total cost of almost £110 it's not cheap).

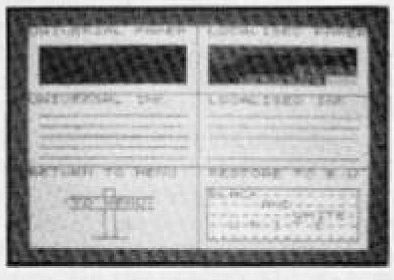
I do have one criticism of *Style* though. The instructions admit that the freehand drawing option is 'virtually useless' when using a joystick. Now I've always considered freehand drawing to be one of the most important facilities offered by graphics utilities (there are, after all, slow but adequate commands for circles and lines in BASIC), and the inability to use this facility without the additional cost of the Graphics Pad strikes me as a rather significant flaw-especially as the other options available are relatively limited when compared to other, cheaper utilites.

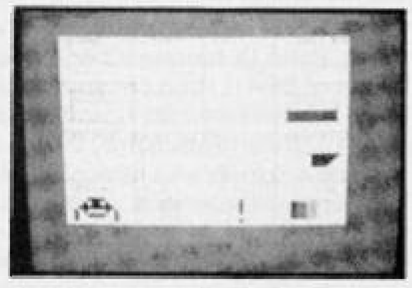
To make the most of *Style* you really need to use it in conjunction with the Graphics Pad, and I must admit that I found using the two together very enjoyable. Admittedly the combined cost of Pad and software together seems high, but you simply won't get any kind of graphics tablet for a lower price. I've noticed that the advertising for *Style* mentions that it is used by software houses and educational establishments, and these are probably the sort of customers that *Style* is primarily aimed at (because they can afford it), and I think that schools in particular would find it genuinely useful. For the home user though, I can only say that I enjoyed using *Style* and that it's at least worth trying out if you're sufficiently interested in graphics to consider the investment.

For further details, Saga can be contacted at 2 Eve Road, Woking, Surrey (04862 22977).









Light Screen Designer



Part Nine: by Toni Baker

ADDRESS	SUBROUTINE POINTED TO		PURPOSE	HOW OBTAINED
E2E1	DD E4	DEFW E4DD,ITAL_OFF	Italics Off	SYM SHF/Q
E2E3	E2 E4	DEFW E4E2,ITAL_HS	Half Slope Italics	SYM SHF/W
E2E5	E8 E4	DEFW E4E8,ITAL_FS	Full Slope Italics	SYM SHF/E
E2E7	F2 E4	DEFW E4F2,TEXT_EXIT	Cancel Text Mode	CAP SHF/SPACE
E2E9	D0 E4	DEFW E4D0,7_BIT	7 Bit Wide Toggle	TRUE VIDEO
E2EB	D4 E4	DEFW E4D4,BOLD	Bold Type Toggle	INVERSE VIDEO
E2ED	3A E3	DEFW E33A,CAPS_LOCK	Caps Lock Toggle	CAPS LOCK
E2EF	52 00	DEFW 0052,RET	Not Used	EDIT
E2F1	53 E4	DEFW E453,TC_LEFT	Cursor Left	CURSOR LEFT
E2F3	45 E4	DEFW E445,TC_RIGHT	Cursor Right	CURSOR RIGHT
E2F5	5F E4	DEFW E45F,TC_DOWN	Cursor Down	CURSOR DOWN
E2F7	72 E4	DEFW E472,TC_UP	Cursor Up	CURSOR UP
E2F9	52 00	DEFW 0052,RET	Not Used	DELETE
E2FB	F2 E4	DEFW E4F2,TEXT_EXIT	Cancel Text Mode	ENTER
E2FD	01 E5	DEFW E501,ALT_SIZE	Alter Chr Size	BOTH SHIFTS
E2FF	43 E3	DEFW E343,GRAPHICS	Graphic Mode Toggle	GRAPHICS
E301	01 00	DEFW 0001	HEIGHT of characters.	
E303	01 00	DEFW 0001	WIDTH of characters.	
E305	FF 81 81 81			
	81 81 81 FF		PIXEL LAYOUT of text-cursor.	

OK. Hands up all those people who spotted that if you used the ARC_RAD routine in the last issue then HIDE and CURSOR_TYPE subsequently didn't work! You didn't notice? Well it's lucky I'm observant then. This bug was caused by a particularly daft oversight on my part, whereby, the ARC_RAD routine corrupted the IX register — (actually it's corrupted by the ROM, but that's neither here nor there). Since the majority of the Light Screen Designer program relies upon IX having a constant value of DB40 then I'm sure you will appreciate the problems that such corruption may cause.

The solution is to re-assign IX within the main loop. If this is done then it will not matter whether the daughter routines corrupt IX, since the mother routine will always restore it. To correct the bug, it is therefore necessary to type in the following BASIC program, along with Light Screen Designer, RUN the BASIC, and re-SAVE the machine code. The correction program is as follows:

```
10 FOR X=0 TO 61
20 POKE 56793+X, PEEK (56797+X)
30 NEXT X
40 POKE 56855,221
50 POKE 56856,33
60 POKE 56857,64
70 POKE 56858,219
80 POKE 56915,23
```

Right — that's over and done with. Now onto this episode. This is a particularly intricate piece of machine code engineering. This is the TEXT option, available on the P key — easy to remember because PRINT is on the same key, and if you've got an old style Spectrum 'Minus' with rubber keys then quotes ('') are on the same key. Once you hit text mode the familiar crosswire cursors disappear, and are replaced by a small rectangle — this is the TEXT MODE CURSOR which you can manipulate around the screen with the cursor keys (with CAPS SHIFT if you have an ordinary Spectrum). But Text Mode allows you to do a whole lot more than just move a different shaped cursor around the screen. Allow me to elucidate.

First of all you could press a number or a letter. If you do this then the required number or letter will appear on the screen where the cursor was, and the cursor itself will move one square to the right. With letters, of course, you can also use CAPS SHIFT to get a capital letter. (Note: CAPS LOCK works as expected). You can also use SYMBOL SHIFT to get a symbol — but this works slightly differently to normal.

If the normal use of SYMBOL SHIFT would be to produce a symbol then this works exactly as it should. If however, it would normally produce a token keyword (for instance SYMBOL SHIFT/Y which would normally produce the keyword 'AND') then you get something different. In general you get the E-Mode symbol on the same key (for instance SYMBOL SHIFT/Y would give square left bracket) — this means that you never have to enter E-Mode, but there are some exceptions. The COPYRIGHT symbol for example. SYMBOL SHIFT/P will produce quotes, so I have taken the liberty of moving COPYRIGHT onto a different key. SYMBOL SHIFT I will now generate the required COPYRIGHT symbol. This leaves just three keys — Q, W and E — which don't have a symbol on them, and these are used as control keys, which will be explained in a moment.

As well as covering letters, numbers, and symbols, you can also use the graphics characters. Just press the GRAPHICS button to enter graphic-mode as you would normally, and the same button

```
                           ORG E30D
FDCB016E  T_CHR      BIT 5,(FLAGS)
28FA                 JR Z,T_CHR          Wait until a key is pressed.
FDCB01AE             RES 5,(FLAGS)       Signal "Ready for a new key".
FD46CE               LD B,(LAST_K)       B:= character code of key pressed.
CD541F               CALL 1F54,BREAK_KEY Test for CAPS SHIFT/SPACE.
3802                 JR C,T_CHR_1        Jump unless CAPS SHIFT/SPACE pressed.
0603                 LD B,03             03 signals CAPS SHIFT/SPACE.
78        T_CHR_1    LD A,B              A:= character code.
FEAA                 CP AA
D8                   RET C               Return if A contains a valid code.
216A02               LD HL,SYMBOL_CODES  Point HL to ROM table of "Symbol
                                         Shift Character Codes".

EDB1                 CPIR                Locate character in table.
11DBFF               LD DE,FFDB
19                   ADD HL,DE           Reposition HL so that it now points
                                         into "Extended Mode with Shift" table.
7E                   LD A,(HL)           A:= E-Mode symbol on same key.
FEBF                 CP "IN"
2002                 JR NZ,T_CHR_2       Jump unless SYM SHIFT/I pressed.
3E7F                 LD A,"©"            COPYRIGHT symbol now on SYM SHF/I.
A7        T_CHR_2    AND A
F0                   RET P               Return with codes 00, 01 or 02 for
D6B5                 SUB B5              SYM SHF/Q, SYM SHF/W and SYM SHF/E
C9                   RET                 respectively.
                     ORG E33A
3A6A5C    CAPS_LOCK  LD A,(FLAGS2)
EE08                 XOR 08
326A5C               LD (FLAGS2),A       Complement CAPS LOCK flag.
C9                   RET
                     ORG E343
3A415C    GRAPHICS   LD A,(MODE)
EE02                 XOR 02
32415C               LD (MODE),A         Complement GRAPHICS flag.
C9                   RET
```

```
                    ORG E34C
C5        T_LOCATE  PUSH BC
FE80                CP 80
3006                JR NC,T_LOC_1       Jump if A contains either a graphic
                                        symbol or a UDG.
ED4B365C            LD BC,(CHARS)       Point BC to Pixel-Layout table in
                                        ROM, less 0100h.
1808                JR T_LOC_2
D690      T_LOC_1   SUB 90
380D                JR C,T_LOC_3        Jump if A contains a graphic code.
ED4B7B5C            LD BC,(UDG)         Point BC to Pixel-Layouts for User
                                        Defined Graphics. (Note: Range of
                                        A is now 00 to 19h).
6F        T_LOC_2   LD L,A
2600                LD H,00             HL:= character code.
29                  ADD HL,HL
29                  ADD HL,HL
29                  ADD HL,HL
09                  ADD HL,BC           HL: points to pixel layout of
                                        required character.
C1                  POP BC
C9                  RET
47        T_LOC_3   LD B,A              Low nibble of B indicates which
                                        graphic symbol is required.
CD380B              CALL 0B38,PO_GR_1   Construct graphic symbol in MEMBOT.
21925C              LD HL,MEMBOT        Point HL to this newly created
                                        pixel layout.
C1                  POP BC
C9                  RET
```

again to get back to normal. In this manner, the keys 1 to 8 will produce block graphics, which may be inversed by the additional use of SHIFT. The letter keys will produce the User Defined Graphics. In this program, five extra UDG's are allowed — on keys V, W, Y and Z. To use them it is necessary to allow an extra forty bytes in the UDG are of RAM, and graphics V, W, X, Y and Z may be defined in this area.

To break with tradition, SPACE will print a space — it will not escape from Text Mode. There are two ways to exit Text Mode. One is to press SPACE with CAPS SHIFT, and the other is to press ENTER.

Characters

It's worth pointing out at this stage that the characters are not necessarily printed on whole character squares. Depending upon the position of the cursor a character may overlap two or more character squares. You see when you hit TEXT MODE and you change from crosswire cursor to text cursor, the text cursor is printed as an outline, whose top left hand corner exactly coincides with the position of the crosswire cursor — to the exact pixel!

```
                    ORG E371
F5        PR_NZ_DE  PUSH AF             Stack the zero flag.
3E20                LD A,"space"
D7                  RST 10              Print a space.
F1                  POP AF              Restore the zero flag.
2007                JR NZ,PR_NZ_1
3E20                LD A,"space"
D7                  RST 10              Print a second space.
3E20                LD A,"space"
D7                  RST 10              Print a third space.
C9                  RET
7A        PR_NZ_1   LD A,D
D7                  RST 10              Print CHR$(D);
7B                  LD A,E
D7                  RST 10              Print CHR$(E);
C9                  RET

                    ORG E384
C5        T_COLOUR  PUSH BC
D5                  PUSH DE
E5                  PUSH HL             Stack registers.
CD0B0B              CALL 0BDB,PO_ATTR   Colour appropriate attribute byte.
E1                  POP HL
D1                  POP DE
C1                  POP BC              Restore registers.
23                  INC HL              HL: points to next byte in screen.
3E08                LD A,08             A:= number of pixels in this byte.
C9                  RET
```

You can print text with this facility in many different ways. We've already seen the effect of CAPS LOCK and GRAPHICS, but what about the rest of the control keys?

TRUE VIDEO will switch between 7-pixel wide, and 8-pixel wide characters. 7-pixel wide characters are obtained by overprinting the 8th column of pixels, which leaves less space between two consecutive characters. This will give you thirty six characters across the screen instead of thirty two. I personally think that this makes text look much better.

INVERSE VIDEO will switch between standard and bold typeface. Bold type is produced by printing each INKEd pixel twice instead of once. Bold type looks very nice, and of course there's nothing to stop you using 7-pixel wide bold characters if you want to.

You can also print characters in *italics*. SYMBOL SHIFT/W will switch to half-slope italics (sloping two pixels down for every one pixel across); SYMBOL SHIFT/E will switch to full-slope italics (sloping at forty-five degrees); and SYMBOL SHIFT/Q will switch to fully upright characters (italics off).

Finally, you can change the *size* of characters. You are not restricted to printing characters the size of a standard character square — you can have double width, double-height, triple height, and so on. To achieve this press both shifts together (or the Extended Mode key on the Spectrum Plus). You can then alter the size of the cursor by pressing the cursor keys: RIGHT to increase the width, and LEFT to decrease the width; DOWN to increase the height, and UP to decrease the height. Spae will allow you to ESCAPE from Extended Mode and return the cursor keys to their normal functions. Any other key will be acted upon as normal, as well as escaping from E-Mode.

```
                    ORG E391
3E08      TX_PRINT  LD A,08
08        TXLOOP_1  EX AF,AF'           A':= Number of rows in pixel layout.
3A01E3              LD A,(HEIGHT)
32675C              LD (B_REG),A        (B_REG):= Number of times each row
                                        must be duplicated.
3A40DB    TXLOOP_2  LD A,(J_FLAGS)
EE10                XOR 10
3240DB              LD (J_FLAGS),A      Complement ODD/EVEN ROW flag.
C5                  PUSH BC             Stack cursor coordinates.
D5                  PUSH DE             Stack pixel layout pointer.
E5                  PUSH HL             Stack address of screen-byte
                                        containing cursor.
79                  LD A,C              A:= column number of cursor (in pixels).
E607                AND 07
4F                  LD C,A              C:= position within screen-byte
                                        of cursor.
1A                  LD A,(DE)
57                  LD D,A              D:= current row from pixel layout.
DDCB005E            BIT 3,(J_FLAGS)
2803                JR Z,TXP_1          Jump unless using BOLD typeface.
87                  ADD A,A             Shift left one bit. Bit 0 reset.
B2                  OR D                Use each bit twice, if set.
57                  LD D,A              D: now ammended for BOLD type.
3A03E3    TXP_1     LD A,(WIDTH)
47                  LD B,A              B:= number of times each pixel
                                        must be duplicated.
3E08                LD A,08             A:= number of pixels in screen-byte.
5F                  LD E,A              E:= number of bits in pixel layout row.
0C                  INC C
1803                JR TXALLIGN2
3D        TXALLIGN1 DEC A               A:= number of remaining pixels in (HL).
CB06                RLC (HL)            Move next pixel from screen into bit 7.
0D        TXALLIGN2 DEC C
20FA                JR NZ,TXALLIGN1     Pixel at cursor position now
                                        in bit 7 of (HL).
C5        TXLOOP_3  PUSH BC             Stack character width. (B).
F5                  PUSH AF             Stack screen-byte pixel counter.
CB02                RLC D               Move next pixel from pixel-layout
                                        into bit 0 of D.
```

Overall then, it would seem that the text facilities of LSD are pretty powerful. You can put text wherever you want it — to the exact pixel — you can print it any size you want, and in a large variety of typefaces (twelve in total). You can escape from text mode by pressing CAPS SHIFT/SPACE or ENTER. You can re-enter text mode by pressing key P, or return to BASIC by pressing SPACE.

The Light Screen Designer series is now very nearly complete. The next issue will contain the penultimate episode, which will be all about drawing ellipses. The final part — part eleven — will do all the colouring in. I know a lot of people have been waiting for that facility. After part eleven there will be an Epilogue (I prefer to call it an Epilogue rather than a Part Twelve because the *program* will be completed in Part Eleven). This will be just to round everything off nicely. I'll leave you now to feed in all this machine code. Bye bye till next time.

Toni Baker

```
FD4E57   TXLOOP_4   LD C,(P_FLAG)         Bit 0 of C:= OVER status.
7E                  LD A,(HL)
07                  RLCA                  Bit 0 of A:= current pixel from screen.
A1                  AND C                 Bit 0 of A:= 0 if OVER 0; or current
                                          pixel from screen if OVER 1.
CB19                RR C
CB19                RR C                  Bit 0 of C:= INVERSE status.
A9                  XOR C                 If INVERSE 1; then complement
                                          bit 0 of A.
AA                  XOR D                 Bit 0 of A:= current pixel from
                                          required pixel-layout, with OVER and
                                          INVERSE correctly implemented.
1F                  RRA                   Carry:= pixel to print.
CB16                RL (HL)               Bit 0 of (HL): correctly stores pixel.
F1                  POP AF                A:= number of remaining pixels in
                                          (HL) to consider.
3D                  DEC A
CC84E3              CALL Z,E384,T_COLOUR  If all 8 bits of (HL) have been dealt
                                          with then colour the attribute byte
                                          and point HL to next byte from screen.
F5                  PUSH AF               Stack screen-byte pixel counter.
10E9                DJNZ TXLOOP_4         Duplicate pixel as many times as
                                          is required.
F1                  POP AF                A:= screen-byte pixel counter.
C1                  POP BC                B:= number of times each pixel must
                                          be duplicated.
1D                  DEC E
20E0                JR NZ,TXLOOP_3        Repeat this proceedure for all pixels
                                          within current pixel-layout row.
FE08                CP 08
2808                JR Z,TXALLIGN4        Jump if HL points to a new byte.
CB06     TXALLIGN3  RLC (HL)              This loop re-alligns (HL)
3D                  DEC A                 so that all pixels are back in
20FB                JR NZ,TXALLIGN3       the right place.
CD84E3              CALL E384,T_COLOUR    Colour in the last attribute.
E1       TXALLIGN4  POP HL                HL: points to first screen byte
                                          affected in current row.
D1                  POP DE                DE: points to current row of
                                          pixel-layout.
C1                  POP BC                B:= current row number.
                                          C:= column number of first pixel
                                          affected in current row.
```

```
CD36DD              CALL DD36,DOWN_PIX    Point HL and B one pixel down.
DDCB0046            BIT 0,(J_FLAGS)
280D                JR Z,TXP_3            Jump unless printing in italics.
DDCB004E            BIT 1,(J_FLAGS)
2004                JR NZ,TXP_2           Jump if using full slope.
DDCB0066            BIT 4,(J_FLAGS)       Test ODD/EVEN ROW flag.
C413DD   TXP_2      CALL NZ,DD13,LEFT_PIX If reqd, point HL and C one pix left.
FD352D   TXP_3      DEC (B_REG)
208C                JR NZ,TXLOOP_2        Repeat current pixel-layout row
                                          as many times as is required.
13                  INC DE                DE: points to next pixel-layout row.
08                  EX AF,AF'
3D                  DEC A
2080                JR NZ,TXLOOP_1        Repeat procedure for all rows
                                          of pixel-layout.
C9                  RET
```

```
                    ORG E414
F5       TX_CURSOR  PUSH AF
C5                  PUSH BC
D5                  PUSH DE
E5                  PUSH HL               Stack all registers.
E5                  PUSH HL               Stack HL again.
2A905C              LD HL,(MASK_T)
E3                  EX (SP),HL            Stack system variables MASK_T and
                                          P_FLAG. HL now restored.
E5                  PUSH HL
21FF01              LD HL,01FF
22905C              LD (MASK_T),HL        Set INK 8; PAPER 8; BRIGHT 8;
                                          FLASH 8; OVER 1; INVERSE 0.
E1                  POP HL
1105E3              LD DE,TC_PIXEL        Point DE to pixel layout for cursor.
CD91E3              CALL E391,TX_PRINT    Print the cursor onto the screen.
E1                  POP HL
22905C              LD (MASK_T),HL        Restore original colours.
E1                  POP HL
D1                  POP DE
C1                  POP BC
F1                  POP AF
C9                  RET

                    ORG E434
3A03E3   GET_WIDTH  LD A,(WIDTH)          A:= width of character (in squares).
D5                  PUSH DE
5F                  LD E,A
87                  ADD A,A
87                  ADD A,A
87                  ADD A,A               A:= 8*(WIDTH)
DDCB0056            BIT 2,(J_FLAGS)
2801                JR Z,GET_WID_1        Jump unless using 7-bit typeface.
93                  SUB E                 In which case A:= 7*(WIDTH).
D1       GET_WID_1  POP DE
C9                  RET
```

```
                    ORG E445
D5       TC_RIGHT   PUSH DE
CD34E4              CALL E434,GET_WIDTH
5F                  LD E,A                E:= character width in pixels.
87                  ADD A,A               A:= twice this width.
81                  ADD A,C
3802                JR C,TCR_1            Jump if cursor right not possible.
93                  SUB E
4F                  LD C,A                C:= new column number of cursor.
D1       TCR_1      POP DE
182C                JR J_PIXADDR
D5       TC_LEFT    PUSH DE
CD34E4              CALL E434,GET_WIDTH
5F                  LD E,A                E:= character width in pixels.
79                  LD A,C                A:= current cursor column number.
93                  SUB E                 A:= new cursor column number.
D1                  POP DE
D8                  RET C                 Return if cursor left not possible.
4F                  LD C,A                C:= new column number of cursor.
1820                JR J_PIXADDR
D5       TC_DOWN    PUSH DE
3A01E3              LD A,(HEIGHT)         A:= character height in squares.
87                  ADD A,A
87                  ADD A,A
87                  ADD A,A
5F                  LD E,A                E:= character height in pixels.
87                  ADD A,A               A:= twice this height.
80                  ADD A,B
FEB0                CP B0
3002                JR NC,TCD_1           Jump if cursor down not possible.
93                  SUB E
47                  LD B,A                B:= new row number of cursor.
D1       TCD_1      POP DE
180D                JR J_PIXADDR
```

```
D5          TC_UP       PUSH DE
3A01E3                  LD A,(HEIGHT)        A:= character height in squares.
87                      ADD A,A
87                      ADD A,A
87                      ADD A,A
5F                      LD E,A               E:= character height in pixels.
78                      LD A,B               A:= current row number of cursor.
93                      SUB E                A:= new row number of cursor.
D1                      POP DE
D8                      RET C                Return if cursor up not possible.
47                      LD B,A               B:= new row number of cursor.
C341DD      J_PIXADDR   JP DD41,PIX_ADDR     Calculate HL:= address of screen-byte
                                             containing cursor, and return.
```

```
                        ORG E482
7E          TEST_CSIZE  LD A,(HL)            A:= character width (or height).
A7                      AND A
2804                    JR Z,TCS_1           Jump if width (or height) zero.
FE10                    CP 10
3802                    JR C,TCS_2           Jump if width (or height) valid.
3601        TCS_1       LD (HL),01           Set width (or height) to 01.
23          TCS_2       INC HL
3600                    LD (HL),00           Reset high byte.
23                      INC HL
C9                      RET
                        ORG E491
E5          TEST_CPOS   PUSH HL
2101E3                  LD HL,HEIGHT
CD82E4                  CALL E482,TEST_CSIZE Ensure valid character height.
CD82E4                  CALL E482,TEST_CSIZE Ensure valid character width.
E1                      POP HL
AF                      XOR A                A:= 00.
DDCB0046                BIT 0,(J_FLAGS)
280C                    JR Z,TCP_1           Jump unless printing in italics.
3A03E3                  LD A,(WIDTH)         A:= character width in squares.
87                      ADD A,A
87                      ADD A,A
DDCB004E                BIT 1,(J_FLAGS)
2801                    JR Z,TCP_1           Jump if using half-slope
87                      ADD A,A
B9          TCP_1       CP C
3801                    JR C,TCP_2           Jump unless current cursor position
                                             overhangs left hand edge of screen.
4F                      LD C,A               Cursor now as far left as possible.
CD34E4      TCP_2       CALL E434,GET_WIDTH  A:= character width in pixels.
81                      ADD A,C
3004                    JR NC,TCP_3          Jump unless current cursor position
                                             overhangs right hand edge of screen.
91                      SUB C
ED44                    NEG
4F                      LD C,A               Cursor now as far right as possible.
3A01E3      TCP_3       LD A,(HEIGHT)        A:= character height in squares.
```

```
87                      ADD A,A
87                      ADD A,A
87                      ADD A,A              A:= character height in pixels.
80                      ADD A,B
FEB0                    CP B0
3806                    JR C,TCP_4           Jump unless current cursor position
                                             overhangs bottom of screen.
90                      SUB B
ED44                    NEG
C6B0                    ADD A,B0
47                      LD B,A               Cursor now as far down as possible.
18AF        TCP_4       JR J_PIXADDR         Calculate HL:= address of screen
                                             byte containing cursor, and return.
```

```
                        ORG E4D0
3E04        7_BIT       LD A,04              Bit 2 of A set.
1802                    JR BOLD_7
3E08        BOLD        LD A,08              Bit 3 of A set.
DDAE00      BOLD_7      XOR (J_FLAGS)        Complement required bit.
3240DB                  LD (J_FLAGS),A
C9                      RET
DDCB0086    ITAL_OFF    RES 0,(J_FLAGS)      Signal "Not printing in italics".
C9                      RET
DDCB008E    ITAL_HS     RES 1,(J_FLAGS)      Signal "Not using full slope".
1804                    JR ITAL_1
DDCB00CE    ITAL_FS     SET 1,(J_FLAGS)      Signal "Using full slope".
DDCB00C6                SET 0,(J_FLAGS)      Signal "Printing in italics".
189F        ITAL_1      JR TEST_CPOS         Adjust cursor position if necessary.
2214DB      TEXT_EXIT   LD (CURSOR),HL       Store final position of cursor.
ED4316DB                LD (CURSOR+2),BC     Store final cursor coordinates.
CD6E0D                  CALL 0D6E,CLS_LOWER  Clear lower part of screen.
F1                      POP AF               Drop return address TEXT_LOOP.
C9                      RET                  Return to LSD Main Loop.
```

```
                        ORG E4FE
CD91E4      ASZ_1       CALL E491,TEST_CPOS  Ensure that cursor will fit on screen.
CD14E4      ALT_SIZE    CALL E414,TX_CURSOR  Print cursor on screen.
FDCB01AE                RES 5,(FLAGS)        Signal "Ready for a new key".
FDCB016E    ASZ_2       BIT 5,(FLAGS)
28FA                    JR Z,ASZ_2           Wait until a key is pressed. Note
                                             that bit 5 of FLAGS remains SET
                                             on exiting this loop.
C5                      PUSH BC              Stack cursor coordinates.
E5                      PUSH HL              Stack cursor address.
CD8E02                  CALL 028E,KEY_SCAN   E:= keyboard code for this key.
E1                      POP HL               HL:= cursor address.
C1                      POP BC               BC:= cursor coordinates.
7B                      LD A,E               A:= keyboard code of key pressed.
CD14E4                  CALL E414,TX_CURSOR  "Undraw" cursor from screen.
FE20                    CP 20
2005                    JR NZ,ASZ_3          Jump unless SPACE pressed.
FDCB01AE                RES 5,(FLAGS)        Signal "Ready for a new key".
C9                      RET                  Return to TEXT_LOOP.
FE04        ASZ_3       CP 04
200B                    JR NZ,ASZ_4          Jump unless 5 pressed with either
                                             shift (CURSOR LEFT).
3A03E3                  LD A,(WIDTH)         A:= width of character.
3D                      DEC A
28D5                    JR Z,ALT_SIZE        Jump back if width=1.
3203E3                  LD (WIDTH),A         Decrement width.
18D0                    JR ALT_SIZE
FE13        ASZ_4       CP 13
200D                    JR NZ,ASZ_5          Jump unless 8 pressed with either
                                             shift (CURSOR RIGHT).
3A03E3                  LD A,(WIDTH)         A:= width of character.
3C                      INC A
FE10                    CP 10
30C4                    JR NC,ALT_SIZE       Jump if width greater than 9.
3203E3                  LD (WIDTH),A         Increment width.
18BC                    JR ASZ_1             Jump back to check position
                                             on screen.
```

```
FE03      ASZ_5     CP 03
200D                JR NZ,ASZ_6          Jump unless 6 pressed with either
                                         shift (CURSOR DOWN).
3A01E3              LD A,(HEIGHT)        A:= height of characters.
3C                  INC A
FE10                CP 10
30B3                JR NC,ALT_SIZE       Jump back if height greater than 9.
3201E3              LD (HEIGHT),A        Increment height.
18AB                JR ASZ_1             Jump back to check position
                                         on screen.
FE0B      ASZ_6     CP 0B
C0                  RET NZ               Return unless 7 pressed with either
                                         shift (CURSOR UP). Note that since
                                         bit 5 of FLAGS is set, the key
                                         pressed will be re-interpretted
                                         within the main TEXT LOOP.
3A01E3              LD A,(HEIGHT)        A:= height of characters.
3D                  DEC A
28A5                JR Z,ALT_SIZE        Jump back if height=1.
3201E3              LD (HEIGHT),A        Decrement height.
18A0                JR ALT_SIZE
```

This is the main entry point for TEXT MODE. When you press the TEXT key (key P) while running the Light Screen Designer control commences from this point. On entry B contains the row number, and C the column number, of the main cursor. HL points to the byte from the screen which contains this cursor.

```
                    ORG E561
CD91E4    TEXT_MODE CALL E491,TEST_CPOS  Move cursor if necessary.
C5                  PUSH BC              Stack coordinates.
E5                  PUSH HL              Stack cursor position.
CDCCDC              CALL MESSAGE
08                  DEFB 08              Print "Now in text mode".
E1                  POP HL
C1                  POP BC
C5        TEXT_LOOP PUSH BC              Stack coordinates.
E5                  PUSH HL              Stack cursor position.
3A3B5C              LD A,(FLAGS)
F5                  PUSH AF              Stack FLAGS (in particular bit 5).
AF                  XOR A                A:= 00.
CD0116              CALL 1601,CHAN_OPEN  Select channel zero (lower screen).
F1                  POP AF
323B5C              LD (FLAGS),A         Restore bit 5 of FLAGS.
011101              LD BC,0111
CD9B0A              CALL 0A9B,AT_BC      PRINT AT 1,17d;
3E4C                LD A,"L"
FDCB305E            BIT 3,(FLAGS2)
2802                JR Z,TEXT_1          Jump unless CAPS LOCK on.
3E43                LD A,"C"
FDCB074E  TEXT_1    BIT 1,(MODE)
2802                JR Z,TEXT_2          Jump unless in GRAPHICS mode.
3E47                LD A,"G"
D7        TEXT_2    RST 10               Print cursor mode.
115348              LD DE,4853           D:= "H", E:= "S".
DDCB004E            BIT 1,(J_FLAGS)
2802                JR Z,TEXT_3          Jump unless using full slope.
1646                LD D,"F"
DDCB0046  TEXT_3    BIT 0,(J_FLAGS)
CD71E3              CALL E371,PR_NZ_DE   Print italic status.
114D43              LD DE,434D           D:= "C", E:= "M"
DDCB0056            BIT 2,(J_FLAGS)
CD71E3              CALL E371,PR_NZ_DE   Print compressed-letter status.
```

```
114F42              LD DE,424F           D:= "B", E:="O".
DDCB005E            BIT 3,(J_FLAGS)
CD71E3              CALL E371,PR_NZ_DE   Print bold status.
E1                  POP HL
C1                  POP BC
CD14E4              CALL E414,TX_CURSOR  Draw t   text cursor.
C5                  PUSH BC
E5                  PUSH HL
CD0DE3              CALL E30D,T_CHR      Wait for key pressed. A:= chr code.
E1                  POP HL
C1                  POP BC
CD14E4              CALL E414,TX_CURSOR  "Undraw" the text cursor.
116CE5              LD DE,TEXT_LOOP
D5                  PUSH DE              Stack TEXT_LOOP as return address.
F5                  PUSH AF              Stack key pressed.
E5                  PUSH HL
C5                  PUSH BC
3E02                LD A,02
CD0116              CALL 1601,CHAN_OPEN  Select channel 2 (Main Screen).
BF                  CP A                 Set the zero flag.
CD5E0D              CALL 0D5E,TEMPS_P    Reset OVER and INVERSE from
                                         permanent colours.
C1                  POP BC
E1                  POP HL
F1                  POP AF               A:= key just pressed.
FE20                CP 20
300F                JR NC,TEXT_4         Jump unless A contains a ctrl code.
E5                  PUSH HL
6F                  LD L,A
2600                LD H,00
11E1E2              LD DE,E2E1
29                  ADD HL,HL
19                  ADD HL,DE            HL: points to subroutine address.
5E                  LD E,(HL)
23                  INC HL
56                  LD D,(HL)            DE:= subroutine address.
E1                  POP HL               Restore HL value.
D5                  PUSH DE
C9                  RET                  Jump to required subroutine.
```

```
C5        TEXT_4    PUSH BC
E5                  PUSH HL
CD4CE3              CALL E34C,T_LOCATE   HL:= address of pixel layout.
EB                  EX DE,HL             DE:= address of pixel layout.
E1                  POP HL
C1                  POP BC
C5                  PUSH BC              Stack cursor coordinates.
CD91E3              CALL E391,TX_PRINT   Print character on screen.
C1                  POP BC               Restore cursor coordinates.
C345E4              JP E445,TC_RIGHT     Move cursor right, if possible,
                                         and return.
```

QL Matchpoint Competition

We've got 20 copies of this ace game to serve up for QL owners!



From Psion, the people who brought you the wondrous QL Chess program (one of the first, and still one of the best games programs for the QL) comes a conversion of their Spectrum game *Matchpoint*.

This tennis arcade/simulation game brings you all the excitement of Wimbledon to brighten up a few dreary winter evenings. Matchpoint allows you to play against the computer or another human opponent, and there's a range of skills and tactics to be mastered that will keep you busy for a long, long time. As you get better, you'll be able to progress through the early rounds of the tournament right through to the finals, and the level of competition gets higher every step of the way!

As soon as we saw Matchpoint running on the QL, we thought it would be a good idea to see if we could arrange a competition of some sort, and, after we made a few approach shots Psion responded by lobbing 20 copies of the game back into our court. Then, just to finish off the rally, they suggested we make a special offer of £1 off the retail price (£14.95) just for readers of ZXC.

So, to enter the competition all you have to do is answer a few questions (about tennis of course) and send your answers in to ZXC. Or, if you want to take advantage of our special offer, then just send payment of £13.95 and the £1 voucher direct to Psion at the address listed on the voucher.

The questions

1) Name the current Wimbledon Men's Single Champion.
2) Who won the Wimbledon Ladies' Championship in 1977 (just as a hint, it was a British player)?
3) Name the current British No. 1 Men's player.
4) Is the Current Wimbledon Women's Champion:
   a) John McEnroe?
   b) Martina Navratilova
   c) Ray Elder?

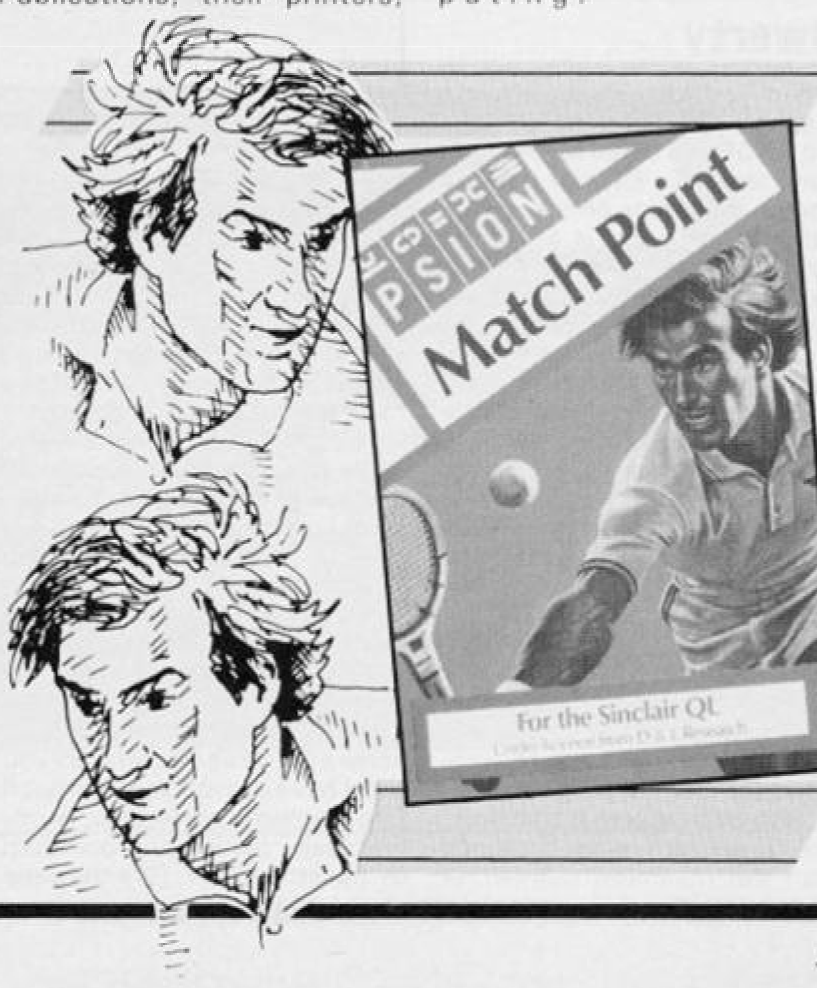
If you've got all the answers, write them on the back of an envelope or postcard and send it to: Matchpoint Competition, ZX Computing, 1 Golden Square, London W1R 3AB.

The rules

- This competition is open to all UK and Northern Ireland readers of ZX Computing, except employees of Argus Specialist Publications, their printers, distributors, employees of Psion Ltd, and anyone else connected with the competition.
- All entries must be postmarked before 31st of January 1986. No correspondence will be entered into with regard to the competition, and it is a condition of entry that the Editor's decision is final.
- As long as each entry is sent written on the back of a separate postcard or envelope, then there is no limit to the number of entries that may be submitted by an individual.
- The winners will be notified by post, and the results published in a future edition of ZX Computing.



Wish I could play as well as the QL



The Perfect Keyboard

John Wase discusses how to customize your keyboard.

It was this lady on the Oxford train. She was telling me how she was involved in a charity which deals with handicapped children's reading problems. 'Yes,' she said 'we've got a Spectrum and like it very much; it's inexpensive and so easy to use'. I thought about that several times in the following week you know, she's right. I suppose that's why I keep adding bits onto mine — essentially it's a basic, inexpensive and rather nice machine, even if you can't type with it. That was my problem when I first wanted to use it as a word-processor. So I seized and read as many add-on keyboard reviews as I could. Some of these are listed below. The list is by no means exhaustive. Several of them mention touch-typing being possible for a number of keyboards, but essentially this is just not true. Let's look why. . .



Figure 1. DK Tronics keyboard

Qwerty . . .

When you learn to touch-type the 'home' keys for the left hand are ASDF and those for the right hand are JKL;. Each of the four fingers thus rests on a key. The index (first finger) is also used for G (left hand) and H (right hand) respectively. To access the rows above and below, the hands move up and a little left or down and a little right.

To convert the keys accessed by the left hand to capitals or equivalent, the shift key at the other end of the board is depressed, using the little finger of the right hand. And vice versa. So, to get any capitals properly, you need a shift key at each end of the bottom line, symmetrically placed, and situated conveniently for the little finger of either hand. Exit most add-on keyboards 'recommend for touch typing'.

However, worse is to follow. The Spectrum was designed with the minimum number of keys. A QWERTY keyboard has semicolon as the 'home' key (little finger, right hand side), and a 'single entry' full stop and comma on the bottom line. These are missing on many add-on keyboards, but it is very difficult to touch type without, at any rate, the full stop and comma.

Finally, because each key has to be used for much more than just lower case and caps, additional symbols are obtained through symbol shift, through extended mode, (caps shift *and* symbol shift), and through symbol-shifted extended mode. A few of these are needed for normal typing, (e.g '!'), and Tasword uses double-shift for some control characters. So you must have a symbol shift at each end, again, preferably symmetrically placed and operated by the little finger, (like the caps shift). Oh, and of course you need a space-bar

Membrane Misery

Well, when I looked, I could find no keyboard that fulfilled all these criteria. So I needed one that I could modify. This cuts down the choice a bit, too! if you don't happen to have a joystick, you will know that over-enthusiastic pounding of the cursor buttons eventually does them in. If you then prise apart the Spectrum case in your fury,you will find the (damaged) membrane. This consists of three plastic sheets glued together. The top one and bottom one have circuits printed in their inner surfaces; the middle one is blank, but the holes cut where the circuits cross, corresponding with the rubber buttons above. So the two outer pieces of membrane are forced together when the button is pressed, making the appropriate contact. This principle is also used in some add-on keyboards, (the Saga Emperor 1 for instance). And, because of this membrane *problem*, they are *very* difficult to modify; to modify a key-board, it must essentially consist of separate keys mounted on a circuit board.

Padding along

As far as I was concerned, I had one additional criterion to meet, I often have to enter quantities of numeric data, either as tables in the text,or as a tabular material for calculations. So, for me a numeric pad was particulary useful.

At the time I was buying, there were far fewer keyboards available than now, and I recall considering two. Transform's was around £70 whilst DK Tronics new space-bar board was around £45 (it's now nearer £30). Both of these have separate keys. Those on the Transform board are of the plated-contact sort whilst those on the DK'Tronics board are of the cheaper individual membrane-deforming type. These are quite good enough for many purposes, and the price was attractive. So I bought a standard DK'Tronics board; this



Figure 2. My first modification.

is probably the most popular add-on keyboard in use

Secretarial scrutiny

The first thing I did was to enlist the aid of our professional office secretaries. I had half a dozen of them pound the keyboard for a while before modification. None of them liked the touch very much, but reckoned they could get used to it. However, their comments on the original arrangement of the keys were unprintable and gave me the confidence to go ahead with my first set of modifications.

Our Editor often mentions how helpful DK' Tronics are. I can also vouch for this. I explained what I wanted to them and they produced a circuit diagram and spare keys, enabling me to fit my single key entry 'full stop' and 'comma' and the extra symbol shifts. The keyboard (first modification) now looked like the second picture (Fig.2).

Full of delight — (yes, bless you Dk'Tronics — it worked!) — I produced my modified keyboard for the secretaries to pound. Pride becomes before a fall! They all gave it the thumbs down. When I got down to it, there were two further problems. The first and major one was that I couldn't count. Look at a standard 'QWERTY' keyboard. On the bottom line, after 'M' and 'N', there's a comma, and a full stop, then at least one more key before the caps shift, (very commonly '?' with '/' when shifted). I had put the caps shift there, hadn't I! Because the caps shift was in the wrong place, they felt very uncomfortable and unfamiliar, and somehow they were all cramped up — several complained that it felt cramped. The second problem is that the 'enter' key is next to the 'L, where one normally has a ';' key, and a couple of others to the right for that matter. And the same on the rows above: the secretaries complained that their fingers 'fell off the end of the keys' — somehow it didn't feel right at all.

So, I set out to modify the board again. Back to DK' Tronics. 'Please sirs, can I now . . . ?' 'No you can't,' they said. 'No matter how many diodes you fit, you won't get the single entry keys you want: can't be done.'

That did it!

Right note, wrong key

The shifted keys on the Spectrum are obtained by first pressing one key and holding whilst holding another. To simulate this without the use of diodes one needs a double-pole switch in which one set of contacts is made before the other, and which matches the originals in shape, size and feel. Armon Electronics, DK'Tronics' suppliers, were very helpful. The first batch of keys they supplied were not double-pole, (although they had thought they were), but did reveal that I had got another unsuspected problem. At sometime DK' Tronics had changed their key pattern, and my board was about the last made with the now-obsolete keys; the ones used are now about 1.5 mm longer. After several months, they managed to get a sample of double-pole keys from Japan, but my personal gremlin was by now having too much fun to give up. Although double-pole, these required far too much pressure to get second contact, (which was right at the bottom of the travel). And these were the only keys Armon could supply.

I then tried National Switch Factors at Keighley, who were also very helpful and sent me several sorts of double-pole keys. However, these all worked rather like hinges, and were



Figure 3. The finished job.

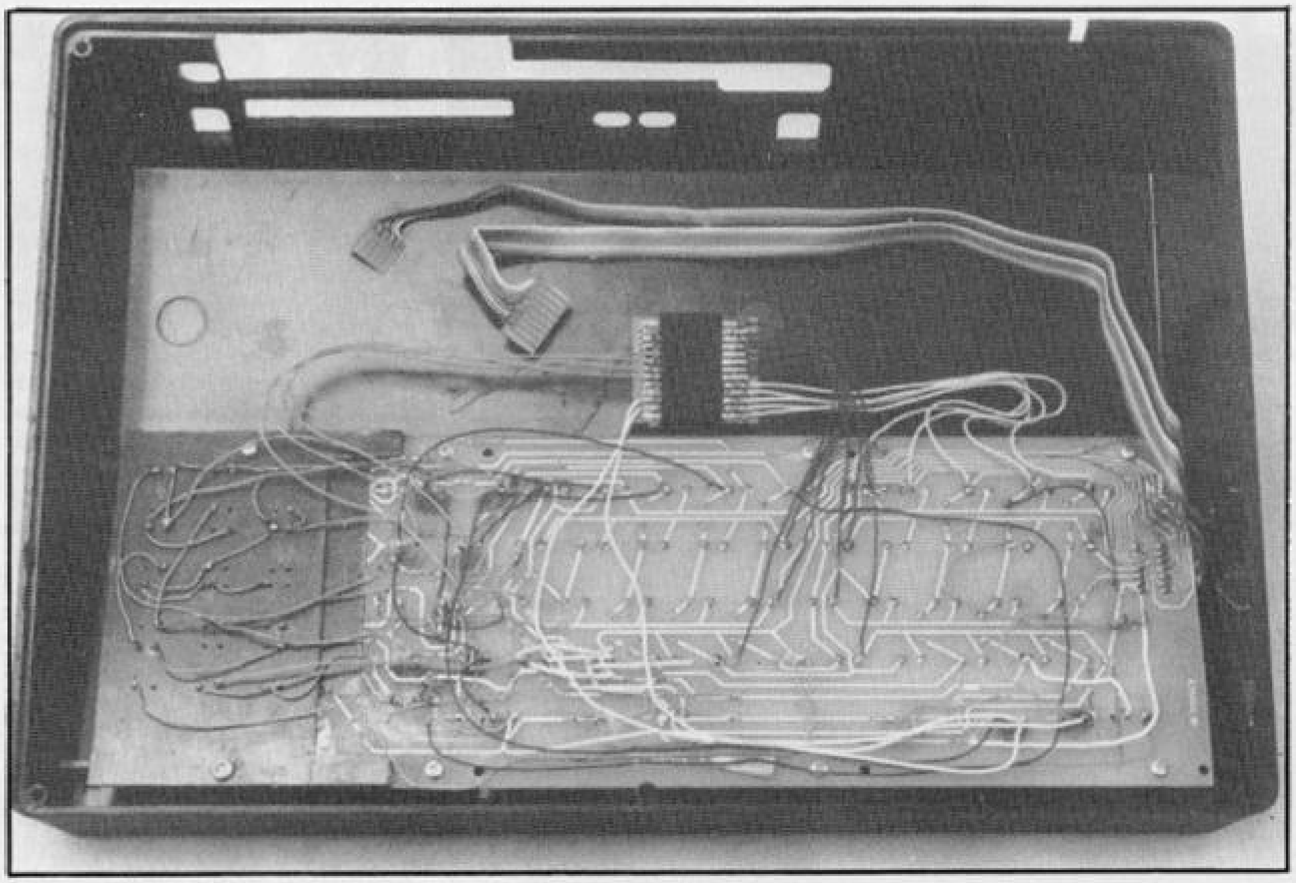


Figure 4. The spider-web inside, connected up to the Kappa interf a c e

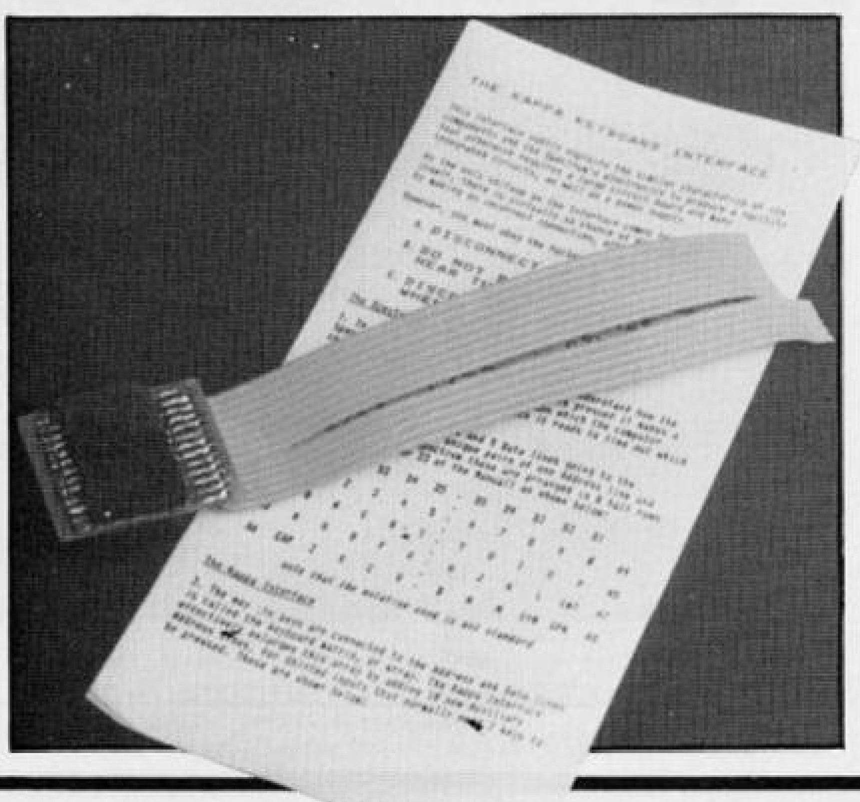
the wrong size and pressure. It looked like the end of the road. What I didn't know at the time was that there *are* keys of the type I wanted available. For instance, more recently, the keyboard which was made by Cheetah, (and is now marketed by LMT), uses this principle. And I suspect that Cherry keyboards also use this type of key for some applications. However, at the time, just as I was desparing, a friend came up with a new approach, and so I abandoned my search for double-pole keyswitches.

Key to the problem

He recommended a little gadget designed by one Steve Drain and marketed by Kappa keyboards. This has a row of terminals on either side. On one side you connect all the input lines into the keyboard. On the other you can connect lines to single key entry symbol-shifted or caps-shifted keys. A diagram and instructions are provided and it is all there, but I found it not particulary easy to follow. Fortunately, I had to visit the Evesham Micro Centre's branch at Cotteridge (famous for there Interface III) and they gave me an old Spectrum membrane of the type mentioned earlier. Tracing the circuits on this helped immeasurably. However, I still needed some extra keys of the old pattern. Back to DK'Tronics. . .

At that stage, I shouldn't have rung up our Editor and promised this article, should I? Dk' Tronics scoured their depot and their keyboard factory, but no luck. Weeks passed; no keys. In desperation, I ordered a dozen of the new pattern. After all, I needed to interpose extra keys where the numeric pad once was, so I had to move it along a bit. I therefore sacrificed it, remade it with the new keys on a spare piece of board and stuck this on under a cut-out on the original board. Thus, the new keys were mounted slightly lower, which just about compensated for their increased height. This was a much easier alternative to grinding all the caps down and released a dozen old pattern keys to use as the extra keys on my modified board (I felt rather pleased about that).

Figure 5. The Kappa kit



So, at last I had the thing together. The DK' Tronics case needed additional carving (fortunately it's very receptive to this sort of ill-treatment, being ABS plastic and therefore relatively easily worked). I then had a major hiccup because it wouldn't work. I contacted Steve Drain who resoldered my connections correctly (he has apparently had bad press recently over some postal problems, but I found him very helpful). And there we were — at last a working keyboard.

Happy endings

A few final comments. It's much easier to have a keyboard to modify rather than to start from scratch. Dk'Tronics proved very easy to alter and has a reliable name. New keyboards are available from a variety of suppliers (e.g. Armon Electronics); alternatively, second hand ones are available. Do make sure that the keys are of the contact variety, not capacitance or Hall effect type and that sufficient spare keys are available. You will have to solder all the data and address lines into the kappa interface, even if you don't need all the outputs. And finally, don't try this device unless you are bold and prepared to use a soldering iron. Happy keyboarding.

Acknowledgements
I am grateful to the help of a number of people, in particular Keith Thompstone of DK'Tronics, Armon Eletronics Ltd., 109, Wembley Hill Road, Wembley, National 'switch Factors, Keighley,and Steve Drain of Kappa keyboards, 14 Pauls Mead, Portland, Dorset, DT5 1JZ.

Literature

Some reviews of add-on keyboards may be found in Your Spectrum (7), Sep 1984, pp28-31; Sinclair Answers, Jan/Feb 1985 pp 4-8; Sinclair User, Jun 1984, pp45-47. Additional information is provided in ZX Computing, Apr/May 1984, pp53-55; Oct/Nov 1984, p18; Feb/Mar 1985, p80; Apr/May 1985, p118: and in Sinclair User, Aug 1984, pp29-31; oct 1984, p23; Dec 1984, pp56, 57, 59; Jan 1985, p25; Feb 1985, p47; Jun 1985, p47; Jul p41; Aug, pp39, 40.

Midi Control of the DX7 and TX7.

We try a Spectrum MIDI package which makes DX7 programming easier, and TX7 editing possible.

At over £1400 we did not consider DX7 software to be of interest to many readers, but from the number of letters we received we realised we were wrong!

The TX7, a DX7 without keyboard, designed to be added to a DX7 and priced at £699, was more feasible but none — including Yamaha it seems, thought it could be programmed without a DX7. But we discovered that it could!

I purchased a TX7 a little while ago. A DX7 was out of my price range, but I reasoned I could link it to the Siel keyboard we used for an earlier review, and this proved to be so. Although the voices supplied programmed into the unit were more than satisfactory, I soon became aware that I only had access to about 60% of the machine's capabilities. So, I rang up Mike Beecher of Electromusic Research and neither he nor the programmer of their DX7 software knew whether it would function with the TX7, although they couldn't see why not. The programs arrived and I set about experimenting.

The programs

Throughout this article I will be referring to the TX7. For DX7 owners ALL the operations and functions of the software are the same and work perfectly — I managed to borrow a DX7 in order to confirm this. The three programs marketed by EMR are:

1. A voice and function library

This consists of seven banks of 32 sounds. These are transferred to and from the DX/TX via the EMR MIDI interface.

This was an immediate bonus to the TX as the only other way of doing this was the cassette interface fitted and, although tape sound libraries probably exist, I was unable to find any.

Now I had access to 224 alternative sounds. The sounds supplied are interesting, ranging from the pure effects such as 'birds' to some beautiful and pleasing musical sounds. I was told that they had been collected from many sources from all over Europe, in themselves they are a valuable addition to the TX/DX. If your use of the equipment is studio or home based then having the Spectrum connected permanently with the full library in memory is satisfactory as loading and finding a soundfrom a bank takes around twenty seconds!

However, if you are using your machine on the road, or want to use your Spectrum to run another MIDI program such as EMR's Performer — a brilliant 8 track real time polyphonic 'tape recorder' simulator, then this is not often convenient. To get around this you can use this program to compile a bank of the 32 most frequently used sounds and keep it in the DX/TX's memory permanently. This is a little time consuming and rather fiddly, requiring loading a bank to the TX, retrieving the single voices required back to the Spectrum, saving onto tape one at a time until all 32 are on tape. Then by loading each voice back into the Spectrum, transferring to the DX/TX and storing into the required memory number, the final compilation of the 32 voices can be made.

2. Function Parameter Store.

A fundamental difference between the TX7 and the DX7 is that the TX7 provides a separate set of function information for each voice, whereas the DX7 has only one which must be changed, if necessary, when a new voice is used. This means that this program is not so essential to TX7 owners especially as each function is alterable from its own control panel. However the easy to use menus plus the means of displaying all the information at once on the screen proved very useful and greatly eased the modification of the values.

A big bonus for DX7 owners is the option to load one of the bank of 32 function patches into the DX7 from the program change of another MIDI instrument, thus saving the need to reprogram them manually. After each alteration of a function the patch is sent to the TX/DX so that you can play it and get an instant audio feedback of what your modification sounds like.

3. Voice Editor.

This is the chief program for DX/TX owners in that is displays all the information clearly and graphically, both in bar graph and line graph form. As the actual programming of the DX/TX voices is well known for its complexity, this program should be received with open arms by those who are still struggling. The fast access to all the parameters which can be seen at a glance, compared and altered, almost makes this task easy!

After each section has been changed the new information is sent to the TX/DX and can be played so you can hear the effect you have created. For DX7 owners this program will aid your sound creation if you are experienced or make it much less daunting to learn how to program sounds.

TX7 Specific problems

Because the TX7 does not have a keyboard and is connected by the MIDI IN to the controlling keyboard, I had a problem in that every time I wanted to use the program I couldn't play the sounds and vice versa. This meant a lot of pulling out and plugging in of the 5 pin Din leads. Not a good idea.

I got around it by purchasing the Yamaha YME-8 MIDI box. This allows two MIDI INs to be switch routed to two banks of four MIDI OUTs. To swap from computer to keyboard was now only a matter of flipping two switches. Expensive at £55 but cheaper than replacing damaged leads and sockets, also it come in useful for other MIDI operations.

Yamaha say in their TX manual — "as the TX7 voices are not programmable there is no need to give details of programming them". This means that you have no clue how to go about creating sounds. If you have a good idea of sound creation on digital synths then you are likely to be able to work it out by trial and error.

Whatever your knowledge it is great fun and satisfying (not to mention frustrating) using the Voice Editor program to experiment with.

Plus...

ZXC has managed to organise a special offer in conjunction with Electromusic Research. Individually, the Function Library, Function Editor, and Voice Editor programs cost £24.95 each — which would make a total of £74.85 for all three. ZXC is able to offer its readers all three programs for the bargain price of just £49.95, representing a saving of er, um (hang on a sec)... £24.90!! (EMR's own Midi Interface is also available for £79.90).

Binary and Hex

The Arithmetic of Computing

by
Carol Brooksbank

It might seem odd to start an article about binary and hex by talking about decimal arithmetic, but the processes involved in all three systems are the same. Only the number of digits used is different. The trouble is that we are all so expert in using decimal counting that we no longer think about the mechanics of what we are doing. Almost everyone can count from 1 to 100 without worrying about the processes involved, and even someone with no head for figures can add 1 to 99 and get the right answer without a second thought about what they did. They just know how to do it. But if you understand what is happening when you count in decimal, you are half way to becoming fluent in binary and hex, so I am going to analyse decimal counting. Please don't skip this bit just because it seems very elementary. You will get to grips with the other systems much more easily if you read on.

Decimal

In decimal arithmetic, there are 10 digits which are always used in the same order when counting:

0 1 2 3 4 5 6 7 8 9

Because of the need to use numbers higher than 9, a column system is used. Imagine the digits arranged in a vertical column. When 9 is reached, we have run out of digits, so a second column is brought in to the left of the first one, and this new column is assumed to hold 0. To progress beyond 9, 1 is added to the amount in the new column, and the original one is returned to 0. This gives the number we call ten.

```
 0
 1
 2
 3
 4
 5
 6
 7
 8
09
10
```

You can then count on, until you reach 9 again in the right hand column, when you repeat the process, adding 1 to the left column and returning the right one to 0.

```
19
20
```

This process can continue until 99 is reached, when you need to bring in another column to the left.

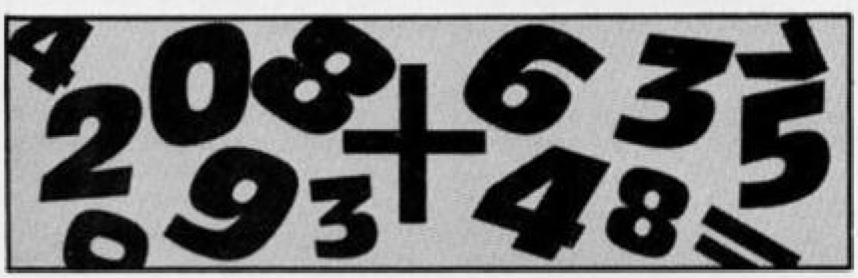
```
 98
099
```

Now, as before, you add 1 to the leftmost column and return all the others to 0.

```
099
100
```

You can go on like this for as long as you need. Every time you add 1 to a column, you return all those to the right of it to 0, and every time all the columns in use reach 9, you bring in another to the left. In this way, any number, as big or small as you wish, can be expressed.

Because there are 10 digits in use, each new column is worth 10 times the one on its right.



The number 548,672 is actually made up like this:

```
100,000 x 5 = 500,000 +
 10,000 x 4 =  40,000 +
  1,000 x 8 =   8,000 +
    100 x 6 =     600 +
     10 x 2 =      20 +
      1 x 7 =       7
      TOTAL    546827
```

This columns system is exactly the same in binary and hex, so the only new factor is the number of digits.

Binary

In binary arithmetic, there are only 2 digits: 0 and 1. As before, you count in one column until you run out of digits, when you introduce a new one to the left. The difference is that you run out of digits much more quickly in binary.

Binary	*Decimal*
0	0
01	1
10	2

The number reached at this point in binary, 10, is not ten. It is one-oh binary. Ten is a name reserved for one-oh decimal, and in order to avoid any confusion, you should always think of the binary numbers as one-oh binary, one-one binary and so on.

You carry on counting in binary in exactly the same way as in decimal, introducing new columns as they are needed.

11	3
100	4
101	5
110	6
111	7
1000	8

You will see by now the great disadvantage of binary. With so few digits the numbers quickly become longer, so that the number expressed as '8' in decimal is a four figure number in binary. In fact, because the numbers are long and in order to reduce confusion between systems, it is usual to express all binary numbers up to 255 decimal as eight figure numbers:

```
  8 decimal = BIN 00001000
255 decimal = BIN 11111111
```

As there are only 2 digits, each column in binary is worth twice the column to its right.

128	64	32	16	8	4	2	1
1	1	1	1	1	1	1	1

To convert a BIN number to decimal, you work as follows:

BIN 11111111 =

```
128 x 1 = 128 +
 64 x 1 =  64 +
 32 x 1 =  32 +
 16 x 1 =  16 +
  8 x 1 =   8 +
  4 x 1 =   4 +
  2 x 1 =   2 +
  1 x 1 =   1 +

 TOTAL 255
```

In binary, just as in decimal, you can introduce as many columns as you need to express a number of any size. Take the decimal number 12345, for instance. In binary, this is

0 0 1 1 0 0 0 0 0 0 1 1 1 0 0 1

Since binary is such a cumbersome system with long awkward numbers, what use is it? Well, in ordinary everyday life, not a lot, but in the world of electronics, it has one great advantage. Because there are only two digits, it is possible to express the numbers electronically.

Imagine that you have a bank of eight electric wires, each equipped with a switch. If you switched the switches so that 0 was represented by an inert wire (switch off), and 1 was represented by a live wire (switch on), you could express the binary form of any number between 0 and 255 decimal. Put very crudely, this is what happens inside your computer. You can think of each address in your computer's memory as a minute bank of 8 electric wires and switches. When you poke a number into an address, the switches are manipulated so that the pattern represents the binary form of your number. The particular combination of live and inert wires causes the computer to perform some operation. It is beyond the scope of this article to go any further into the workings of the computer, but I want you to understand the importance of binary arithmetic

to the computer world. Whatever form of counting you use to program your computer, decimal, hex or binary, it is the binary form which is important, and the binary form is the one stored in the memory. We use the other forms because binary is so unwieldly for we humans, so we let the computer do the conversion.

As each address, or byte, in the computer's memory can only hold 8 binary digits, bits, what happens if you wish to store a number like 12345 decimal, which has more than 8 bits? The computer takes the 16 bit form of the number, and cuts it in half ie 0011000000111001 becomes 00110000001110 01. It then stores each half separately. The second half, the low byte, is stored first, and the first half, the high byte, in the following address.

This is where decimal, which is more familiar to humans, becomes awkward to use in relation to computers. The low byte, BIN 00111001, is 57 decimal and the high byte, BIN 00110000, is 48 decimal. At first glance there seems to be very little connection between 57 and 48 and 12345. In fact, you can find the decimal by multiplying the high byte by 256 and adding the low byte.

256 x 48	= 12288
ADD	57
TOTAL	12345

This lack of an obvious connection between a number and its low and high bytes in decimal is one of the reasons why many machine code programmers prefer to work in hex, which is less unwieldly than binary and more convenient than decimal for programming. But before I go on to talk about hex, I am going to give you some homework. If you plan to do any machine code programming, you will save yourself a lot of looking up in tables if you learn by heart the binary forms of the decimal numbers 0 to 15, so try to memorise this little table.

BIN	DECIMAL
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	10
1011	11
1100	12
1101	13
1110	14
1111	15

Hex

In hex you have 16 digits — 0,1,2,3,4,5,6,7,8,9,A,B,C,D, E,F. The column system for counting is used exactly as before, but instead of bringing in a new column after 9, as in decimal, you carry on until you reach F, and then introduce the new column.

HEX	DECIMAL
0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
A	10
B	11
C	12
D	13
E	14
0F	15
10	16

Again, the number reached in hex is not ten, but one-oh hex, and is equal to 16 decimal. Hex numbers are expressed as 10H or 0010H.

Each column in hex is worth 16 decimal times its right hand neighbour. The number 12345 decimal is 3039h, and is broken down as follows:

4096	256	16	1
3	0	3	9

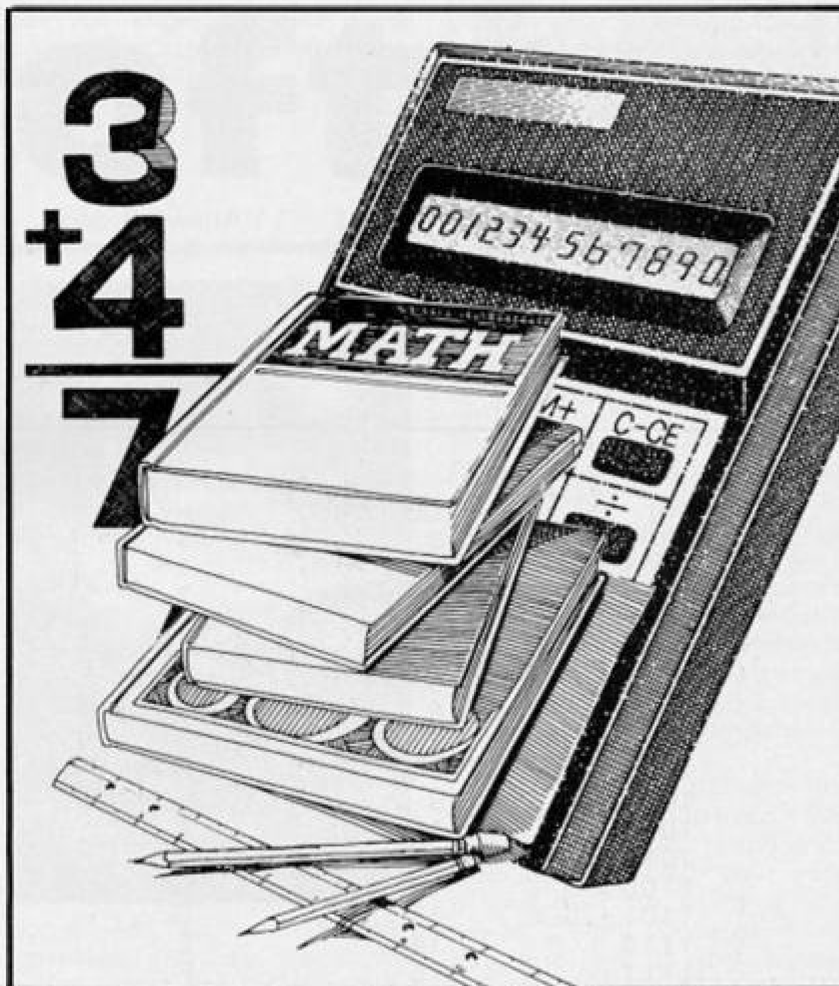
4096 x 3 =	12288 +
16 x 3 =	48 +
1 x 9 =	9
TOTAL	12345

A hex number containing some of the letter digits can be converted as follows:

4096	256	16	1
2	A	C	F

4096 x 2 =	8192 +
256 x 10 =	2560 +
16 x 12 =	192 +
1 x 15 =	15
TOTAL	10959

You will remember that we discovered earlier that the high and low bytes of 12345 were 48 and 57 respectively. We have since seen that the hex form of 12345 is 3039h. The hex form of 48d is 30h and 57d



is 39h. You can see at a glance what the high and low bytes of a hex number are, without doing any calculations. From this, it follows that there is a direct connection between the binary and hex forms of a number which is much more obvious than the connection between the decimal and any other form. (See figure 1).

Figure 1. Binary, hex and decimal comparisons

BIN 0011	= 03h	= 3d
BIN 0000	= 00h	= 0d
BIN 00110000	= 30h	= 48d
BIN 0011	= 03h	= 3d
BIN 1001	= 09h	= 9d
BIN 00111001	= 39h	= 57d
BIN 0011000000111000	= 3039h	= 12345d

Clearly, this makes life much easier for the machine code programmer. By using hex he can avoid all the calculations required to find the high and low bytes when working in decimal. Also, when he comes to type in the machine code listing, most of the numbers are shorter than the equivalent decimal ones so there is a saving in time too. In fact, after you have become familiar with hex you begin to wonder why decimal is the universally popular system. I suspect that is has something to do with the fact that we are all born with a built in decimal calculator — four fingers and a thumb on each hand — and everyone, including primitive man who invented counting, learns to count on their fingers. If we had three extra fingers per hand I believe that we should all be working away happily in hex, and the genius who suggested that you could have a very good counting system with only ten digits would be quietly led away by men in white coats.

Meanwhile, computer programmers need to familiarize themselves with binary and hex. You simply need to become as expert with binary and hex as you are with decimal, and practice is the only way. You can help yourself by learning another conversion table (figure 2). It is very similar to the first one, but will allow you to convert between the three systems. In fact, you have very little extra to learn because decimal and hex figures are the same until you reach 10 decimal.

Can you do sums in hex? ▶

Yes, of course, and in exactly the same way that you do them in decimal. Let us add 3039h and 7/FE3h.

```
 3039
7/FE3
=====
```

9 + 3 = C (try not to work in decimal ie 9 + 3 = 12, convert it to hex = C. Count it out in hex – 9 A B C. You may need help at first, and fingers are not much use as you haven't enough, so write the hex digits out and count along them).

```
3039
7FE3
   C
```

3 + E = 11 Answer 1 and carry 1.

```
3039
7FE3
  1C
 1
```

0 + F = F + carry 1 = 10

```
3039
7FE3
 01C
1
```

3 + 7 = A + carry 1 = B

```
3039
7FE3
B01C
```

Figure 2

BIN	DECIMAL	HEX
0000	0	0
0001	1	1
0010	2	2
0011	3	3
0100	4	4
0101	5	5
0110	6	6
0111	7	7
1000	8	8
1001	9	9
1010	10	A
1011	11	B
1100	12	C
1101	13	D
1110	14	E
1111	15	F

I cannot make you an expert in hex just by writing this article. It will depend on how much time and practice you are willing to put in. Try devising sums for yourself. You might even try writing the multiplication tables in hex. I will start you off with the 'two times' – you can work out the others.

2 x 1 = 2
2 x 2 = 4
2 x 3 = 6
2 x 4 = 8
2 x 5 = A
2 x 6 = C
2 x 7 = E
2 x 8 = 10
2 x 9 = 12
2 x A = 14
2 x B = 16
2 x C = 18
2 x D = 1A
2 x E = 1C
2 x F = 1E

Try to get into the habit of doing all your calculations in hex, rather than working things out in decimal and then converting the answer. It is rather like learning a foreign language. You cannot become fluent if you work everything out in English and then translate it. You have to learn to think in the language. You have to learn to think in hex.

This is why I have not included a hex-decimal conversion table in this article. If you start to rely on one you will never really be able to work freely in hex and it will always be a foreign language. It is much better to persevere with learning to count and calculate in hex, even if you flounder a bit at first. For those who find they simply cannot manage without a conversion table, you will find one in almost any book on machine code programming. But you will know that you have become fluent in hex on the day you no longer feel you need one. ■

Philips Data Recorder

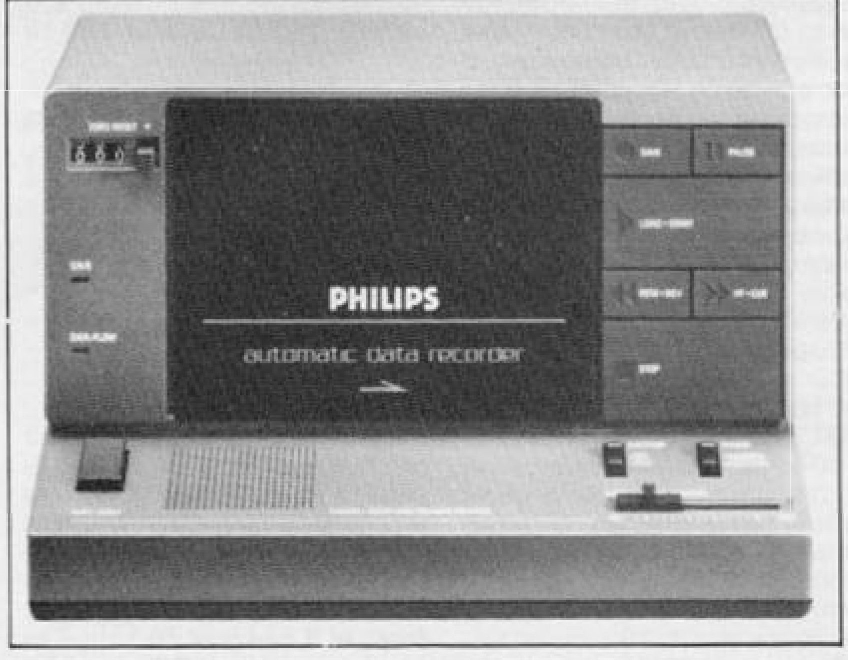
Those nice people from Philips sent us one of their new D6450 Data Recorders to look at recently, and very nice it looks too. The trouble was that when we got set to try it out on a few programs we realised that there's no way of plugging it directly into the mains. You can put half a dozen batteries into it, but that's bound to become expensive after a while (unless you use rechargeable batteries), and you've also got the added hassle of making sure that the batteries are always fully charged or you'll get loading problems.

Your only other alternative is to go out and buy a separate 9v DC power supply which will cost you an additional £10 or so. Considering that the recorder will cost you about £32 on its own, that extra expense ought not to be necessary just to be able to plug it into the mains.

Still, after ripping up an old Oric PSU we got the recorder working and it SAVEd and LOADed all sorts of things without any problems, and it's quite well designed so it wasn't at all fiddly to use. My only criticism is that the need to buy batteries and/or a power supply makes it more expensive than it really needs to be.

If you're interested try contacting Philips Consumer Advice, 420 London Road, Croydon, CR9 3QR (01-689-2166).

Cliff Joseph



■

Quicksoft

Desert Burner
Creative Sparks
£2.50

As leader of a small band of freedom fighters your task is to deliver the plans of a new weapon to another underground group. Your means of transport is a powerful motorcycle called 'Desert Burner' which is capable of over 200mph.

By itself this is a pretty simple task, but what the instructions do not tell you is that you have to emulate Evil Knievel in jumping over various obstacles whilst avoiding bombs dropped by helicopters. You can attack the helicopters by firing at them via the space key. Controls are simple, Caps Shift to accelerate and Z to jump.

As with many games of this simplicity it has an addictiveness out of proportion to the action, it may not be particularly sophisticated, a few flickers, a few attribute clashes, but it is pretty good value for the money. A nice touch is the slowly sinking sun and the moon appearing at night. A high score is kept, but no hall of fame.

GRAPHICS	* * * *
ADDICTIVENESS	* * *
OVERALL	* * * *

Supersam
Budgie Software
£2.50

An unusual plot here. The object is to explore the 34 rooms of a fortress and collect the 18 parts of a cage in which to catch the elusive Boris. This is complicated by misleading doorways which send you astray and potholes which are fatal. Added to this you have an energy level which drops on contact with one of the wandering beasties, when completely depleted you lose one of your three lives. On the positive side are syringes which make you invulnerable for a brief period.

Options for joystick or to redefine keys are included and it features one of the most interesting loading systems I've yet seen. Again, this game has a charm of its own and I happily played away on it, the graphics are fairly good although slight flickering and attribute clashes are evident. The screens are well designed and provide a good challenge. Though essentially a maze/dodge and collect game it is reasonable value for money and will provide a few hours of amusement.

GRAPHICS	* * *
ADDICTIVENESS	* * *
OVERALL	* * *

Nonterraqueous
Mastertronic
£1.99

This game is one which puts some full priced offerings to shame. With over 1000 screens of well designed problems to negotiate, fast action, smooth well animated sprites and a set of problems to solve, this game is a must for any arcade addict.

Put simply it is a variation on the maze search and dodge theme, but objects have not merely to be collected but also used to enable you to get further into the game. The ultimate objective is to explore the inside of a giant mountain to find and then destroy the computer which controls the population of your planet. Options for various joystick interfaces are provided and keys QAPO give up/down/left/right movement. Space fires and I and U allow specific actions. The game is very addictive and the inclusion of a high score chart encourages you to keep trying to improve.

Would I buy it? Yes!

GRAPHICS	* * * * *
ADDICTIVENESS	* * * *
OVERALL	* * * * *

Cylu
Firebird
£2.50

Well you can't avoid comparisons and this one just asks to be likened to Knight Lore. Your quest is to search the maze (yes, yet another maze) find, collect and return to the master computer, 24 objects. To achieve this you will have to solve the problem of how to disable 32



force fields, use the nine teleport terminals and use the master refuel and master return to base units.

This is a game to get your fingers in a twist as there are no fewer that 11 control options plus save/load/abort game options. Movement, graphics and screen designs are very good, bu the game is hard to get into, the problem of running out of fuel caused many a promising game to end rather too soon.

This is one for the expert gamester and should give many hours of enjoyment, however I must admit in the time I had to review it I did not get too far, once the pressure of writing this is off I will be going back for another session.

GRAPHICS	* * * *
ADDICTIVENESS	* * *
OVERALL	* * * *

Chinese Patience
Atlantis
£1.99

I remember Atlantis' card game program *Eights* which I still play on occasion and so was pleased to receive this program. Chinese Patience is another card game and matches you against the computer, and boy does the machine play a mean game!

The instructions take up the whole of the inside of the insert plus the flap, reading them makes the game seem very complicated. However once I



ran through the game a couple of times I got the hang of it and found it frustratingly absorbing.

The graphics are well designed and the screen layout is clear and uncluttered, the program is well error trapped — try pressing break! There is a fairly lengthy wait while the computer makes its moves but no so long that you get bored. I found I took much longer than the computer to decide on my moves.

If you have the patience and like card games (and even if you don't I suggest you ty this one), then this will keep your mind ticking over for many a long winter's night. Unusual in this day and age is the loading system, using LOAD " " CODE is a bit amateurish, but this is only my criticism.

GRAPHICS	* * * *
ADDICTIVENESS	* * * *
OVERALL	* * * *

Action Biker
Mastertronic
£1.99

In conjunction with a well known snack manufacturer the character of Clumsy Colin is brought to the computer screen. Using a split graphics and text window you have to ride around town, find Marti and take him to the Spaceport. Control is UP/DOWN/LEFT/RIGHT by keys AZNM plus space to enter a house.

Graphics are OK, movement and animation is fair and the action fast and furious. High scores are kept but no hall of fame. This is a fairly average game which gave me a few interesting moments but no real desire to go back to it again. I know how game players vary in what they like and many of you will enjoy this game, so I would simply suggest you try it out at your local computer store and make up your own mind. It didn't do a lot for me.

GRAPHICS	* * *
ADDICTIVENESS	* *
OVERALL	* * *

Speakeasy

Kai Webber has come up with a machine code routine that turns the Spectrum into a speech synthesiser.

This is by no means the first Spectrum speech program but it does offer far greater manipulation of speech data than some of the other efforts that have appeared in magazines, allowing crackles and noise to be cut down greatly. Basically, what the program allows you to do is to record speech from a tape recorder (along with a few extra noises thrown in for good measure!) and then replay this speech in your own programs.

Begin by carefully typing in the BASIC listing and then save it to tape with SAVE "SPEAKWELL" LINE 1. When you run the program, you should be confronted with some data details at the bottom of the screen which are fairly self explanatory and a menu of options:

R — RECORD SPEECH

This allows you to record your voice into the computer's memory at the current record/play speed. When you select this option you will be told to press any key to start recording. There are two methods you can use to record speech (experiment with both methods to see which gives best results):

1) Record your voice onto a tape and play this tape into the computer, pressing a key just before the speech starts. (Try different volume settings until you achieve the best results).

2) Set the recorder into Record mode (remove any tapes in the recorder, find the small switch in it which checks to see if a tape has a write protect tab, press this in and press 'PLAY' and 'RECORD' on the recorder at the same time), press a key and start talking into the tape recorder's microphone.

P — PLAY BACK SPEECH

This will simply replay any speech stored in memory at the current record/play speed.

A — ALTER START ADDR

This allows you to alter the start address from which speech is recorded/played back. If there is a lot of 'noise' from the tape recorder at the start of some speech you can get rid of it by altering the start address by trial and error. (If the start address plus the length of data would cause the data to overflow into the graphics area of the computer will prevent you from using this value — lower the length first — see below. It will also reject any value below 32855).

L — CHANGE LENGTH

Allows you to change the length of speech to be recorded/played back. The computer will prevent you from using any value which will cause an overflow into the graphics area.

S — CHANGE SPEED

Lets you change the record/play back speed. By playing back at a different speed you can alter the pitch of your voice. The slower the speed, the worse speech will sound. (Values below about 25 are of little use).

M — MEMORY MANAGER

A useful tool that allows you to move blocks of speech around in memory. Three parameters are required: address to copy from; address to copy to and number of bytes to copy. If 'Invalid address' appears on the screen, then you have tried to overwrite either the graphics area of the machine code. If you wish to remove 'noise' that was recorded before you started speaking, find the start address of the actual speech (using 'S') and enter this as the address to copy from. Enter the address to copy to and the number of bytes to copy and then alter the start address to the value you used as the address to copy to. (Got that?!!).

T — TAPE OPERATIONS

This prints up a self-explanatory tape menu. For those of you who are interested, I have included an assembly language listing of the machine-code.

You may find that if you use an expensive tape recorder you will get better results than a cheap portable recorder, but do not do anything silly like feeding the output of a power amp staight into the back of your Spectrum! It also helps if you stress the words as you speak into the tape recorder.

```
  10 BORDER 0: PAPER 0: INK 7: C
LEAR 32767: RESTORE 9920: GO SUB
 9850
  20 LET SPEED=1: LET START=3285
5: LET LENGTH=32512
  30 GO SUB 9730
  40 RESTORE 9720: LET K$=INKEY$
  50 FOR F=1 TO 7
  60 READ A$,LINE: IF A$=K$ THEN
  GO SUB LINE: GO SUB 9810
  70 NEXT F
  80 GO TO 40
9230 PRINT AT 17,0; FLASH 1;"L -
 LOAD    S - SAVE    R - RETURN":
PAUSE 1: PAUSE 0: PRINT AT 17,0,
,
9240 IF INKEY$="l" THEN  GO TO 9
280
9250 IF INKEY$="s" THEN  GO TO 9
320
9260 IF INKEY$="r" THEN  PAUSE 1
: RETURN
9270 GO TO 9230
9280 INPUT "NAME OF FILE?  ";N$
9290 IF LEN N$>10 THEN  PRINT AT
 17,7; FLASH 1;"INVALID FILE NAM
E": GO TO 9280
9300 PRINT AT 17,0,,AT 17,11; FL
ASH 1;"LOADING...": LOAD N$CODE
9310 RETURN
9320 INPUT "NAME OF FILE?  ";N$
9330 IF LEN N$>10 OR N$="" THEN
 PRINT AT 17,10; FLASH 1;"INVALI
D NAME": GO TO 9320
9340 PRINT AT 17,0,,: SAVE N$COD
E START,LENGTH
```

```
9350 PRINT AT 17,10; FLASH 1;"VE
RIFY (y/n)": PAUSE 0: GO SUB 981
0: IF INKEY$="n" THEN  RETURN
9360 PRINT AT 17,10; FLASH 1;"VE
RIFYING..": VERIFY N$CODE
9370 RETURN
9380 INPUT "ADDRESS TO COPY FROM
?  ";HL
9385 PRINT AT 17,0,,
9390 IF HL<32855 OR HL>65366 THE
N  PRINT AT 17,6; FLASH 1;"INVAL
ID COPY ADDRESS": GO TO 9380
9400 INPUT "ADDRESS TO  COPY TO?
  ";DE
9405 PRINT AT 17,0,,
9410 IF DE<32855 OR DE>65366 THE
N  PRINT AT 17,2; FLASH 1;"INVAL
ID DESTINATION ADDRESS": GO TO 9
400
9420 INPUT "NUMBER OF BYTES TO C
OPY?  ";BC
9430 IF BC+DE>65367 OR BC+HL>653
67 THEN  PRINT AT 17,9; FLASH 1;
"INVALID NUMBER": GO TO 9420
9440 POKE 65521,DE-256*INT (DE/2
56): POKE 65522,INT (DE/256)
9450 POKE 65524,BC-256*INT (BC/2
56): POKE 65525,INT (BC/256)
9460 POKE 65527,HL-256*INT (HL/2
56): POKE 65528,INT (HL/256)
9480 IF BC=0 THEN  RETURN
9490 RANDOMIZE USR 65520
9500 RETURN
9510 INPUT "NEW SPEED?  ";SPEED
9520 IF SPEED<0 OR SPEED>255 THE
N  PRINT AT 17,9; FLASH 1;"INVAL
ID SPEED": GO TO 9510
9540 POKE 32854,SPEED
9550 RETURN
9560 INPUT "NEW LENGTH?  ";LENGT
H
9570 IF LENGTH<1 OR LENGTH+START
>65367 THEN  PRINT AT 17,9; FLAS
H 1;"INVALID LENGTH": GO TO 9560
9590 POKE 32852,LENGTH-256*INT (
LENGTH/256): POKE 32853,INT (LEN
GTH/256)
9600 RETURN
9610 INPUT "NEW START ADDRESS?
";START
9620 IF START<32855 OR START>653
60 OR START+LENGTH>65367 THEN  P
RINT AT 17,8; FLASH 1;"INVALID A
DDRESS": GO TO 9610
9640 POKE 32850,START-256*INT (S
TART/256): POKE 32851,INT (START
/256)
9650 RETURN
9660 RANDOMIZE USR 32808
9670 RETURN
9680 PRINT AT 17,0; FLASH 1;"PRE
SS ANY KEY TO START RECORDING":
PAUSE 1: PAUSE 0: PRINT AT 17,0,
,AT 17,11; FLASH 1;"RECORDING"
9690 OUT 254,15: RANDOMIZE USR 3
2768
9700 BEEP 1,5
9710 RETURN
9720 DATA "r",9680,"p",9660,"a",
9610,"l",9560,"s",9510,"m",9380,
"t",9230
9730 PRINT TAB 11;"SPEAKWELL"
9740 PRINT TAB 11;"---------"
9750 PRINT 'TAB 6;"R - RECORD SP
EECH"
9760 PRINT 'TAB 6;"P - PLAY BACK
 SPEECH"
9770 PRINT 'TAB 6;"A - ALTER STA
RT ADDR"
9780 PRINT 'TAB 6;"L - CHANGE LE
NGTH"
9790 PRINT 'TAB 6;"S - CHANGE SP
EED"
9800 PRINT 'TAB 6;"M - MEMORY MA
NAGER"''TAB 6;"T - TAPE OPERATIO
NS"
9810 PRINT AT 19,0;"START ADDR O
F SPEECH DATA:";START
9820 PRINT "END ADDRESS OF SPEEC
H DATA:";START+LENGTH
9830 PRINT "LENGTH OF SPEECH DAT
A:";LENGTH;"     ";#1;AT 0,0;"REC
ORD/PLAY SPEED:";SPEED;"    "
9840 PRINT AT 17,0,,: RETURN
9850 FOR F=32768 TO 32854
9860 READ A: POKE F,A
9870 NEXT F
9890 FOR F=65520 TO 65531
9900 READ A: POKE F,A
9910 NEXT F: RETURN
9920 DATA 243,42,82,128,237,91,8
4,128,6,8,14,0,237
9930 DATA 67,0,0,175,219,254,23,
23,203,17,16,247,113,58
9940 DATA 86,128,71,16,254,35,12
2,179,27,32,226,251,201,243,42
9950 DATA 82,128,237,91,84,128,6
,8,78,203,9,203,9,203,9
9960 DATA 121,230,16,211,254,203
,1,16,247,58,86,128,71,16,254
9970 DATA 203,135,35,122,179,27,
32,224,251,201,87,128,0,127,1
9980 REM BLOCK MOVE MACHINE CODE
9990 DATA 17,0,0,1,0,0,33,0,0,23
7,176,201
```

ASSEMBLY LISTING

```
10  ;         RECORD MODE
20            ORG 32768
30            DI                   ;DISABLE MASKABLE INTERRUPT
40            LD   HL,(START)      ;START ADDRESS OF DATA INTO HL
50            LD   DE,(LENGTH)     ;DATA LENGTH INTO DE
60  LOOP      LD   B,8             ;B CONTROLS LOOP OF 8 BITS/1 BYTE
70            LD   C,0             ;C COUNTS NO. OF BITS AT EAR PORT
80            LD   (0),BC          ;TIME WASTING INSTRUCTION
90  LOOP_2    XOR  A               ;CLEAR A AND CARRY FLAG
100           IN   A,(254)         ;READ PORT 254
110           RLA                  ;VALUE AT EAR PORT...
120           RLA                  ;ROTATED...
130           RL   C               ;INTO REGISTER C
140           DJNZ LOOP_2          ;GET 8 BITS FROM EAR PORT INTO C
150           LD   (HL),C          ;STORE THE 8 BITS IN (HL)
160           LD   A,(SPEED)       ;RECORD/PLAY SPEED INTO A
170           LD   B,A             ;USE B TO CONTROL THE LOOP
180 WAIT      DJNZ WAIT            ;WAIT A SHORT WHILE
190           INC  HL              ;NEXT LOCATION FOR NEXT 8 BITS
200           LD   A,D             ;CHECK WHETHER OR NOT...
210           OR   E               ;TO FETCH ANOTHER 8 BITS
220           DEC  DE              ;DECREMENT THE COUNT
230           JR   NZ,LOOP         ;LOOP IF COUNT NOT ZERO
240           EI                   ;RE-ENABLE THE MASKABLE INTERRUPT...
250           RET                  ;FOR RETURN TO BASIC
260
270
280 ;         PLAYBACK
290           DI
300           LD   HL,(START)
310           LD   DE,(LENGTH)
320 P_LOOP    LD   B,8             ;B CONTROLS LOOP OF 8 BITS/1 BYTE
330           LD   C,(HL)          ;GET DATA BYTE TO BE PLAYED
340           RRC  C               ;GET THE BITS IN THE RIGHT ORDER
350           RRC  C
360           RRC  C
370 P_LP_2    LD   A,C             ;GET THE DATA BYTE INTO A
380           AND  %00010000       ;MASK THE ONLY REQUIRED BIT
390           OUT  (254),A         ;OUTPUT THIS BIT TO THE SPEAKER
400           RLC  C               ;GET NEXT BIT INTO REQUIRED POSITION
410           DJNZ P_LP_2          ;OUTPUT 8 BITS (1 BYTE)
420           LD   A,(SPEED)       ;SPEED INTO A
430           LD   B,A             ;TRANSFER TO B FOR PAUSE
440 PAUSE     DJNZ PAUSE           ;WAIT A SHORT WHILE
450           RES  0,A             ;ANOTHER TIME WASTING INSTRUCTION
460           INC  HL              ;INCREMENT DATA ADDRESS
470           LD   A,D             ;CHECK TO SEE IF COUNT IS ZERO YET
480           OR   E
490           DEC  DE              ;DECREASE THE COUNT BY 1
500           JR   NZ,P_LOOP       ;LOOP IF COUNT NOT ZERO
510           EI                   ;RE-ENABLE MASKABLE INTERRUPT...
520           RET                  ;FOR RETURN TO BASIC
530 START     DEFW 32855           ;START ADDRESS OF SPEECH DATA
540 LENGTH    DEFW 32512           ;LENGTH OF SPEECH DATA
550 SPEED     DEFB 1               ;RECORD/PLAY SPEED
```

Mindplay

This month's 'Mindplay' features three adventure games from the new generation of mass produced adventures written with the aid of Gilsoft's *Quill*, and released by small software companies — Fridaysoft and Sentient. While the Quill is a very useful utility for use by people who wish to write their own adventure I feel that the use of the Quill by software houses may have adverse effects on the adventure game genre. With the Quill it is a lot easier to produce *bad* adventure games, as the thought and planning needed for a non-Quilled game is often not applied. Thankfully, there are exceptions, and these exceptions are often very good games, but the vast majority of Quilled games which are marketed are often average, and sometimes very poor. The games reviewed this issue feature a couple of these exceptions.

You may have noticed a couple of letters in past issues of *ZX Computing* requesting hints and tips for adventure games. Well, in response to popular demand, I plan to introduce clues to adventures in this column. The idea is that you send your adventure problems and they'll be printed in 'Mindplay', and anyone who can help then responds with the answer. Offers of help on adventures will also be printed. In this way adventurers can get in touch with each other through offering or needing help. I'd also be interested in your letters on the general adventure scene. Send all correspondence to Mindplay at the *ZX Computing* address, 1 Golden Square, London W1R 3AB.

Now on with the reviews . . . .

Brian Robb



London Adventure
Fridaysoft
£4.95

The first of this selection of Quilled games is Fridaysoft's *London Adventure* and it is the best of the three adventures reviewed here. It is also the only one to feature a loading screen, a nicely drawn picture of London's Tower Bridge.

The aim of the game is to find the combination of a safe deposit box belonging to your eccentric, rich uncle. He has recently died and his will has left his considerably fortune to you — if you can find it, that is! The combination numbers are hiden within the adventure — for instance you may find a five pound note, so five is one of the combination numbers. The numbers are cleverly hiden within the game, often in the oddest of places!

London Adventure is designed to be realistic (allowing some artistic licence, of course), logical and, apart from the odd diversion and obligatory mazes, geographically correct. The game features over 100 locations including many well known landmarks such as Harrods, Cleopatra's Needle and the Tower of London.

The location descriptions are very good in places, such as aboard the Cutty Sark, near the beginning, while the majority of the descriptions are above average. Some of the locations though, appear to be there only for show, but at many of them you can find objects and clues. On this point it is wise to collect *every* object, no matter how unlikely, as it will no doubt be useful at some point.

The game has quite a few original and amusing touches which raise the adventure above the mass of Quilled games. London Zoo, for example, becomes a particularly clever maze, as do the London Sewers. Another nice touch are the events in the Tower of London — you could lose your head over the Crown Jewels! Also, typing your name on the computer in the Science Museum can be alarming!

Overall, *London Adventure* is a very enjoyable and challenging game which makes a change from the usual fantasy settings, and this is Fridaysoft's avowed intention. You could certainly do worse than buy this one!

Scoop
Sentient Software
£2.99

This adventure, like the previous one, is unusual in its setting and aim. You are an intrepid newspaper reporter and must search the streets for the front page news which will distinguish you from the other hacks. Fail in your task and youll be fired!

The game begins with the piercing sound of your morning alarm. Sound features quite often in the game with the sounds of lifts and car engines, not forgetting the telephone which rings, providing you with a message which moves you forward in the game. You awake, right at the start, to be faced by a morning domestic routine which becomes your first major task — you must get dressed and head off to work. This is a lot more complicated and confusing than you may think — even the simplest tasks are difficult to achieve first time round. Everything must be done in a particular order.

When you eventually find the newspaper office, and this could take a while, the newspaper boss will give you a story to investigate and from there you're on your own! Do you have the intuition which will take you to the top of your profession?

This is a very interesting adventure, with fairly good location descriptions and often involved tasks which are difficult to complete. Like *London Adventure Scoop* is a very good example of a Quilled game at it's best.

The Amulet
Sentient Software
£2.99

This one also comes from Sentient Software, but unfortunately it does not aspire to the novelty of the previous game. Whereas *Scoop* was an inspired and novel game, *The Amulet* repeats the tired and worn fantasy theme of the quest.

The simple aim is to find the seven shards of the shattered Amulet, which are scattered through time. Trying to retrieve the segments entails travelling through time and space using the magical power of the Amulet. You start out with a section of the broken Amulet, which when rubbed enables you to travel in time, in search of the other pieces. Each 'Time Zone' consists of several very similar locations, all with very bland location descriptions. Objects are sparse, so it is easy to find each section of the amulet with very few major obstacles to be overcome.

Some of the locations insist on playing lengthy tunes after each input which are simply irritating and slow down play, particularly the zone featuring the 'Frenchman' (who insists on killing you if you're not careful).

The Amulet is a rather clichéd, and nondescript adventure, with lacklustre location descriptions, devoid of atmosphere and lacking colour.

London Adventure, Fridaysoft, Unit F, The Maltings, Station Rd, Sawbridgeworth, Herts.

Scoop and The Amulet, Sentient Software, Branch House, 18 Branch Rd, Armley, Leeds LS12 3AQ.

We, the jury . . .



Daley Thompson's Supertest Ocean £6.95

The very title of this one is enough to send the most robust of joysticks on strike and make the keyboard tremble in fear — Yep, its another left/right-as-fast-as-you-can test of finger or wrist stamina!

I must admit that I did not like this type of game at first but then it grew on me until now I am hooked. The success of a game of this nature depends on the quality and realism of the graphics, no-one is going to work up a sweat over moving an indistinct blob around, so how have Ocean done? Well I'm sure owners of DT's Decathlon would have no doubts, the graphics are just as good if not better than before and Ocean should have another well deserved hit on their hands.

Even though all the games have only left/right and sometimes fire as controls there is enough variation and judgement required to prevent them from becoming tedious. A brief run down and description of the events on side one is:

PISTOL SHOOTING, move cursor around the screen to random target and fire at centre.
CYCLING, L/R as fast as possible.
SPRING BOARD DIVING, 3 bounces and L/R for somersaults.
GIANT SLALOM, steer L/R through the flags as they scroll up the screen toward you.
ROWING, L/R as fast as you can.
PENALTIES, L/R for speed, line up on football, fire to shoot.
SKI JUMP, L/R for speed, fire to jump and land.
TUG O'WAR, L/R for power.

Each one needs slightly different techniques to succeed and I personally enjoyed the shooting and ski jump whilst the tug O'war and diving gave me a lot of frustration.

GRAPHICS * * * * *
ADDICTIVENESS * * * *
OVERALL * * * * *

Fighting Warrior Melbourne House £7.95

There has been a spate of fighting programs lately, not the least being MH's WAY OF THE EXPLODING FIST, but now they have combined the pure combat game with the adventure quest type to produce this game.

In Fighting Warrior your task is to rescue the princess from a selection of 'vicious' enemies — must be a combination of thick and gluey and brutal.

Joking at the expense of the spelling in MH's press release aside, there is nothing gluey about the beautifully drawn and animated characters in this program, they all act and react very quickly indeed. Your character has to walk along as the background scrolls behind him and every few feet it seems encounters another adversary. Fighting is done by sword and by using the four direction and the fire button, joystick or keyboard you can back off, move forward, jump, duck, or strike at the upper, mid or lower regions of your opponent. Each of you has a graphic representation of your strength and stamina, once it is all gone — RIP.

Added to this there are objects which you have to puzzle out how to use and a frequent arrow attack which you must duck under or jump over, even in the middle of an encounter.

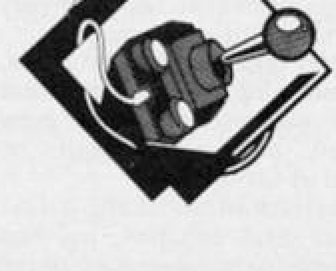
I found that the response was a little slow and it takes some getting used to before accurate timing of blows and dodges can be made. It is also a very difficult game to play as you are under almost constant attack from the very start. Even though I did not get very far into the gameplay, the superb graphics and adventure style concept ensure that I will go back to this one time and again. Melbourne House yet again bring a touch of class to computer gaming.

GRAPHICS * * * * *
ADDICTIVENESS * * * * *
OVERALL * * * * *

















That's The Spirit
The Edge
£7.95

Hello, I thought, a clone of Ghostbusters? But no, although the theme of ghostly beings is the same, in reality there is really no similarity. The game comes complete with the Edge's usual type of unhelpful and mildly humorous instructions on the reverse of which is an overlay for the old type Spectrums which shows which keys perform which operations.

There are 28 actions which you can perform and most of these are fairly clear, Take, Drop, Enter etc. but how and when to use them are left for you to discover. It is with games like these that I wish the company would send us poor reviewers a hint sheet, I tried and I tried, but no matter what I did I only succeeded in being driven insane — the game's end condition — in a few moves.

The screen is a well presented two dimensional representation of New York, and your character moves around meeting with various problems and objects of which the greatest are the ghosts. Not state of the art graphics but very good nonetheless. Animation is smooth although a little on the slow side, and there are some nice touches of humour (try the 'swear' key).

This is a game for the cerebral arcade player and as such is quite difficult, personally I found it a trifle too difficult to get into and so do not feel particularly inspired to play it again.

GRAPHICS	* * *
ADDICTIVENESS	* *
OVERALL	* * *



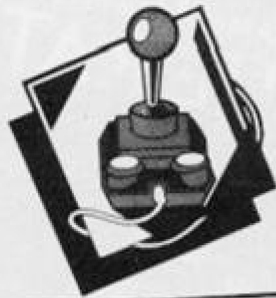
Monty On The Run
Gremlin Graphics
£8.95

Another in the Monty Mole series. In this game he has to escape by travelling through many devious and delightful screens of pure frustration. It seems too simple to describe it as another platform game, but it is. Added to the usual problem of carefully timed jumps is the sneaky device of having to choose and take withyou five items of equipment from a selection of 21 objects before you begin an attempt. Get one wrong and you find yourself stuck and unable to get any further.

The screens are well designed to provide just the right level of difficulty, the animation is good — I loved Monty's somersaulting jumps — and key response is very good.

My only moan is that collision detection seems to be by attributes and this often gives the impression that you have avoided some malignant sprite only to find Monty expiring in front of you. With 43 screens this should keep his fans happy and probably gain him a few new ones. One of the better programs of this type.

GRAPHICS	* * * *
ADDICTIVENESS	* * * * *
OVERALL	* * * *





Bounty Bob Strikes Back
US Gold
£7.95

I'm afraid we only got a review copy of this game and the aim and action was not clearly stated, although they claim it compares favourably with the C64 version.

It is a platform game in which it appears that you have to collect various objects within a time limit, each time you get an object the nasties change from yellow to green for a short time and can be eaten/killed by your character.

Loads of screens, most of which are well designed and present a challenge, some have a nice 3D perspective effect.

I must admit I found the game to be a little inconsistent in that I could clear some screens with ease and others were a real headache, they did not seem to progress logically. Graphics and animation were very good, BB's demise is quite interesting, and the high score hall of fame routine is almost worth playing on its own.

Colourful, enjoyable, I just wish I knew what I was supposed to be doing.

GRAPHICS	* * * *
ADDICTIVENESS	* * *
OVERALL	* * *





Macadam Bumper
PSS
£6.95

It is hard to simulate the old pin table, bumpers, buzzers, bleepers, nudges and all the paraphenalia of the arcade on a home computer. I love pinball tables and have awaited a good computer version for some time.

This one is as close as I think it is possible to get, all the usual features are implemented and you can almost feel the balls bouncing around, giving them just that gentle nudge at the right time. Not only is the standard program very playable but there is also the option to design your own pinball table layout and play it. This is a simple matter although to get the best results a little planning is required at first. The screen is split into two, vertically, and the playing area is on the left while a typical pinball picture is on the right. This also shows scores and players turns etc. In design mode this picture is replaced by the graphic components of the table and they are selected and placed on the playing area as required.

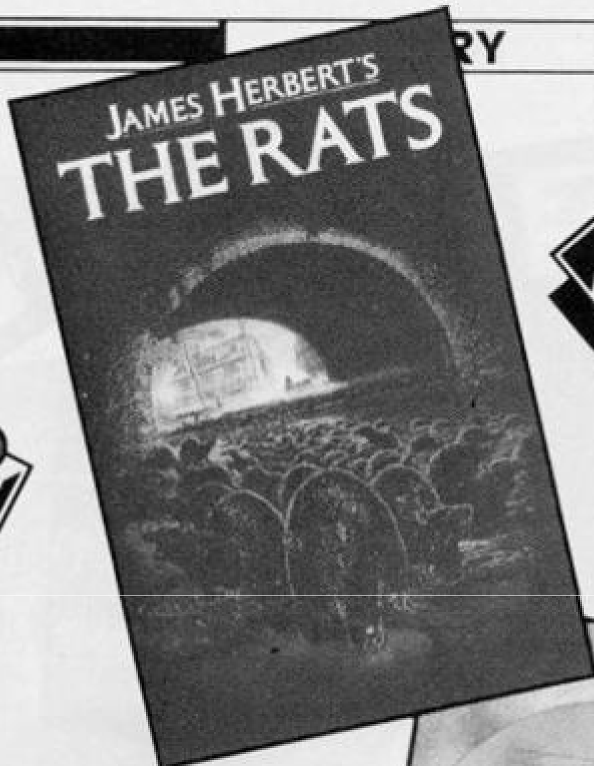
Once designed and completed you can save your work to tape, you can even send it to PSS as an entry to their competition and win your very own real pinball machine. Not bad. As a pinball program it is very playable, and with the option of making your own designs the life of this program is extended greatly. Get bored, design a new game.

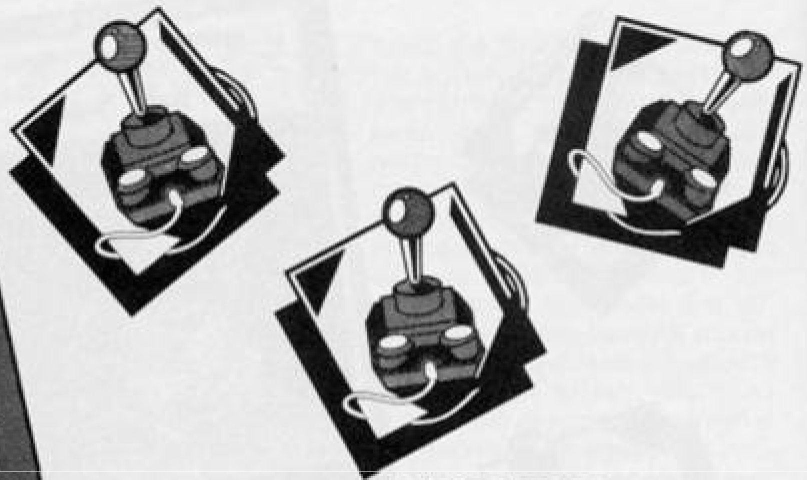
I must admit I enjoyed the fact that I didn't keep losing a game when an overenthusiastic nudge would usually produce a 'tilt' message, this feature does not seem to have been included. However I keep getting the annoying feeling that I've cheated in some way. . .

GRAPHICS	* * * *
ADDICTIVENESS	* * * * *
OVERALL	* * * *

▶







The Rats
Hodder and Stoughton
£7.95

An interesting program which combines strategy and adventure by the use of multiple choice input. The game is based on the James Herbert novel and it does help if you have read this book, however it is not essential and the game can be enjoyed on its own merits.

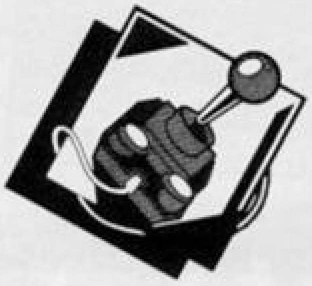
Your prime task is to deploy your forces around London to contain and defeat the rats, this is done by moving a cursor around a map of the city and viewing appropriate sections. Once a section has been chosen, information on the number of men and their equipment is displayed. You can then modify this.

Every now and then the game changes into adventure mode where one of your characters, either one of the main four or one of the many supporting characters, is faced with a meeting with the revolting rodents. In this mode the text input is built up word by word from a bank of options.

Graphics are used to good effect, when you make a fatal mistake the rat leaps through the text. The whole thing is quite an unusual concept and I enjoyed it greatly, however the vivid descriptions are quite gory and not for the squeamish. It will take quite a bit of playing to master this game, especially as the secondary characters load in a random fashion making plenty of game play variation.

This latter feature is not clearly explained and caused a couple of aborted loadings as I thought the program had crashed. If a block is not loading then leave it, it is probably simply not required for that variation of the game.

GRAPHICS ****
ADDICTIVENESS ****
OVERALL *****





Marsport
Gargoyle Games
£9.95

Gargoyle Games have now forsaken Celtic mythology and ventured into space for their latest game. However, Marsport does retain the style of the earlier Tir Na Nog and Dun Darch, in that the game puts you in control of a large animated figure (named John Marsh) who moves around a large playing area, has to solve various puzzles and deal with many creatures in order to complete his task.

In this instance, that task is to save the Earth from conquest by an alien race. You must control John Marsh as he finds his way through a large and complex space station, in order to reach a force field generator that will help to defend the Earth.

The basic style of animation is much the same as Gargoyle's earlier games, but this time the movement of the main character is even smoother than before. The background graphics — depicting the corridors, lifts, and various entrances in the space station — are more complicated than those of Darach/Nog and scroll slightly less smoothly as a result, but even so the overall effect is excellent.

The mechanics of the game have also been enhanced, and Marsport is more complex than its predecessors in some ways, because you now have a greater ability to manipulate objects and engage in combat than before, although trading this off against the memory available has meant that there seem to be fewer independent characters to deal with. Mind you, that's not a criticism as I think I prefer the slightly greater emphasis that this puts on the action elements of the game. Dun Darach was an extremely engrossing game, but couldn't equal the surprise that Marsport will give you when you walk straight into the head baddy.

Marsport is the first in a planned trilogy of games, and if this game is anything to go by I'll be looking forward to the rest of the trilogy.

GRAPHICS *****
ADDICTIVENESS *****
OVERALL *****

Astroclone
Hewson Consultants
£7.95

The Seidabb are back! Hewson Consultants have resurrected their old arch-nasties for their latest game, but this time around the author, Steve Turner has used the same sort of programming techniques that worked so well in his two Avalon games.

Astroclone has a number of similarities to Marsport, in that Hewson have taken a formula that worked well in mythological settings and turned it to a science fiction setting. Taking the part of a crack Astroclone warrior, you must gain control of the Seidabb bases in order to deactivate the Stargates that they used to reach Earth.

The graphic layout of the game is of course similar to that of the Avalon games, but this time you have more than one stage to the game. You can either move around the bases (or Technibo as they are called) in search of the Graviton devices you will need, or you can take off in your fighter and head off to combat the Seidabb cruisers. These two parts of the game provide a nice balance; the action in the bases is of the problem solving arcade-adventure style, whilst the space combat is a fun piece of shoot-'em-up action.

Astroclone is quite complex (the reviewers' notes that Hewsons sent out with the game make quite a fat folder), and though I enjoyed it I'm not sure if it will appeal to everyone. Like Avalon and Dragontorc the problem solving can slow the game down quite a lot in places, and despite the combat sequences the game may not appeal to the shoot-'em-up brigades.

GRAPHICS ****
ADDICTIVENESS ****
OVERALL ****



Sparkler's Special
Creative Sparks
£7.95

Creative Sparks have joined the compilation brigade and lumped together four of their old titles as a sort of software Christmas stocking.

The four titles, which are held in a two cassette package, are; Orc Attack, Stage Coach, River Rescue and Tower of Evil. They're all quite old titles, and none of them were really state-of-the art even when they were new. Orc Attack is probably the most addictive of the games, in which you control a little knight who is at the top of a castle wall and being besieged by hordes of Orcs. The Orcs have got little ladders with which they attempt to scale the walls, and you must fend them off with rocks, sword and burning oil. The screens aren't very varied, but it is one of those simple games that can become quite frantic as the skill levels get harder.

Tower of Evil was a forerunner to Atic Atac in which you have to guide your character through the rooms of a tower, collecting treasure and keys to other rooms. It's mildly enjoyable, but unsophisticated compared to more recent games.

Stage Coach seems to be unplayable, since, try as I might, I couldn't get my cowboy figure to jump successfully from his horse onto the stage coach in order to try and collect the passengers dotted along the route.

River Rescue is a very simple game which allows you to guide a speedboat along a river as you attempt to collect passengers and avoid the alligators and logs bobbing about in the water. The scrolling river doesn't have much variation in detail, and it gets to be a bit of a strain on the eyes after a while.

All in all, though I'm not particularly impressed with any of the games, they do seem the sort of thing that might appeal to younger children who aren't yet up to games as complex as Marsport, so the Sparklers' Special might make a reasonable present for youngsters around the age of ten.

GRAPHICS	* * *
ADDICTIVENESS	* *
OVERALL	* * *



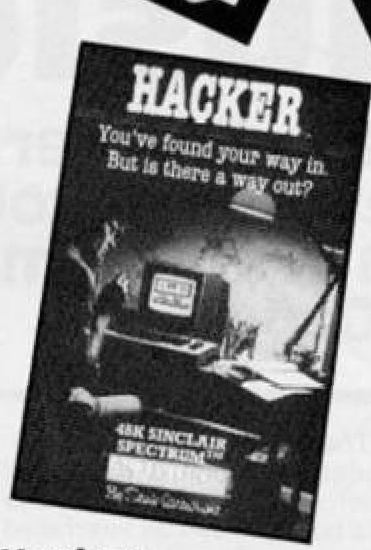
The Secret Diary of Adrian Mole
Mosaic
£9.95

This is a bit of a disappointment I'm afraid, despite being programmed by Level 9. The trouble is that there's not really much of a game here. The program boasts some 200K of text, but most of it seems to have been lifted straight out of the Adrian Mole diaries and is presented onscreen just as if you were reading the books themselves. There is very little opportunity to become actively involved in the game, and, as far as I can tell, nothing of the sort of problem solving that is the essence of most adventures.

User participation takes the form of making decisions for Adrian in his attempts to become as popular as he can. The trouble is that this decision making process is very limited; you are presented with three options (along the lines of 'Shall I read this poem to Nigel/Pandora /the teacher?') and you simply press keys 1-3 as appropriate. There is no opportunity for any independent actions or direct text entry, and you can only sit there reading great chunks of text while you wait for the program to ask you to press a button.

Admittedly, the text is very amusing in places, but if all you want is to read the diaries, then why not go out and buy the books? Buying both the Adrian Mole books would still cost less than this program, they won't give you a stiff neck from staring at the computer screen and they're funnier as well.

GRAPHICS	* * *
ADDICTIVENESS	* *
OVERALL	* *



Hacker
Activision
£7.99

How do you review a game like this without spoiling it for any potential buyers? The whole point about Hacker is that it simulates the experience of breaking into a large corporation's mainframe nd becoming involved in its operations. However, just as you would experience this in reality, there are no instructions to help you determine what is happening or what you are supposed to be doing. In other words, you stumble in 'blind'.

Consequently, there's not much I can say about the game without giving it all away. This isn't a game where you simply have a task to solve — well, yes you do actually, but first of all you have to figure out what that task is. What I can say about Hacker is that it will take you all over the world and involves a search for various pieces of information.

Hacker begins with a simple blue screen, blank except for the words: 'Log On Please', and from here on it's up to you. If you succeed in logging on you'll get into the mainframe and have a to sort out the information that is presented to you.

This is not a game that's likely to appeal to arcade addicts, but for adventure fans and people who like to get their teeth into a good, infuriating puzzle it's probably the best thing to come along since Level 9 adventures.

GRAPHICS	* * *
ADDICTIVENESS	* * * *
OVERALL	* * * *

Halley's Comet
Anima
£8.95

With Halley's Comet just around the galactic corner there seem to be a number of comet related programs popping up at the moment. This one from Anima Scientific Computing actually consists of two programs, a Planetarium which gives a view of the sky and the progress of the comet during both night and day, and an 'Orrery' which is a model of the solar system and the comet's path through it.

The Orrery initially gives a view of the sun and plots the orbits of the inner planets around it. Then, using the cursor keys you can rotate this view allowing you to observe the solar system from any angle, which creates quite a fascinating effect. You can also use the 'V' and 'Z' keys to enlarge or shrink the view to include the orbits of the outer planets as well.

The Planetarium is less interesing visually, but perhaps more useful, since by showing you the path of the comet in the sky it will allow you to trace the comet (and the other planets in the sky) when it does arrive. There are options here to take into account atmospheric effects and to see below the horizon, and, as with the Orrery, to rotate the view so that you are looking at the sky from different directions.

Both programs have options which allow you to manipulate the time scale — either to travel backwards and forwards in time to other appearances of the comet, or to speed up/slow down the passage of the comet as shown on the screen.

Obviously an astronomy program like this will appeal only to a small specialist audience, but it does seem to achieve its aims quite well. It's a little expensive, but after all, Halley's Comet only turns up every seventy years or so, if you're interested why not contact Anima at; 33 Lilac Walk, Tyne and Wear NE31 2LT (0632 832825). ■

The Sound Of Music

John Ainslo presents Interface I owners with some new sound commands for controlling a Programmable Sound Generator.

The Spectrum is a poor relation when it comes to making music or even simple noises to enhance games, as it only has a very simple single channel BEEP command and no noise generator, though by machine code techniques it can be made to create noise.

The Programmable Sounf Generators (PSGs), AY-3-8910 and AY-3-8912, by contrast, offer three tone channels and one noise generator, which can be mixed with any of the three tone channels but these too suffer from one serious drawback in that programming of the PSG is long, tedious and difficult as it requires a minimum of six OUT commands in basic to make even the simplest tone emanate from the loudspeaker.

A variety of sounds which can be incorporated into games to improve them and make them more realistic can be produced by the PSG. The sounds can range from a simple tone to complex mixtures of three channels, with or without noise, which are completely under software control. The PSG also has the ability to continue making its sound while the microprocessor performs other tasks and, unlike the Spectrum's bleeper, can also generate chords.

Fig. 1 shows the block diagram of the 8912 which differs slightly from the 8910 in that it has one In/Out port where the 8910 has two. There are three independent tone generators and a noise generator which can be mixed with a tone or tones, or it can be output on its own. The amplitude of tone channel can be set independently to any one of fifteen values. The envelope generator amplitude modulates the outputs and can be set for fast or slow decay or attack, repeat or single shot.

These two devices were intended for use with CPU's which use a shared address and data bus, but can be interfaced with the Spectrum quite easily. Register selection and data transfers are both done via the data lines and the device is set up to receive either by the state of two control lines BDIR and BC1. All that is required is to select a register by setting both control lines high and then sending the data to that register with BDIR remaining high. The decoding is greatly simplified on the Spectrum by using the IORQ signal together with the required PORT ADDRESS. Thus OUT 221,X will prepare the PSG for data transfer to register X and OUT 223,Y will transfer data Y to that register. The WR, IORQ, A6 and A1 are decoded to give BDIR and BC1. IN 221 can be used to read the contents of the last register pointed to by an OUT instruction.

With the advent of Interface 1 this can all be changed as the user is no longer confined to using IN and OUT commands and complete control of the PSG can now be achieved in EXTENDED BASIC.

This article outlines the two phases of the project. Firstly the method of creating routines to enlarge the normal basic is described in detail, as the principle can be used to create any command required. Secondly, a description with examples, on how to use the newly created commands to try to maximise their usefulness.

The New Commands

Three new commands have been created to allow the programming of the PSG which will be syntax checked on entry and run like the normal commands in that their use will be transparent to the user, though the means of achieving this is entirely different. The three new commands are:-

*SOUND
*MUSIC
*PLAY

but there is no limit to the number of new commands that can be added by the same method.

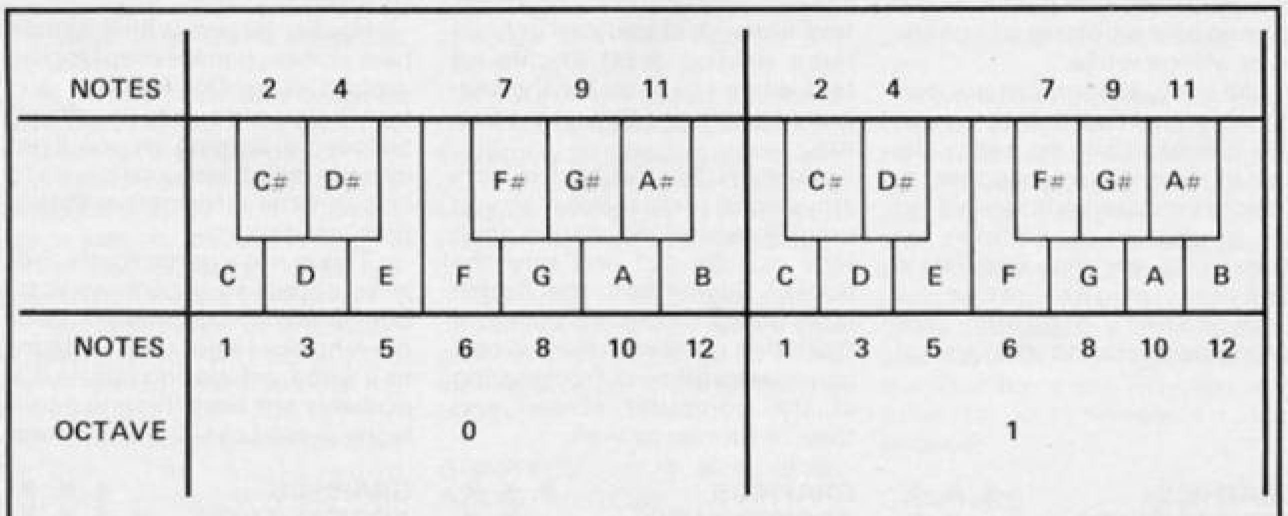
The new commands are machine code routines stored in RAM and the code, once loaded, can be initialised with one RAND USR call, which will point the error vector to the new routine and programming can them begin.

Extended Interpreter

When a BASIC line is entered, the old ROM checks it for syntax errors. If this test fails, further checks are done by the Interface 1 ROM which now checks the line for Interface 1 commands such as 'MOVE', which gave an error message in the unexpanded system but will now be accepted. If these syntax checks also fail, the processor jumps to the error handling routine at ERR 6 via the address vector created with the new system variables at 232735 and 23736. These addresses normally point to 01F0H but can be re-directed to the new routines where further checks can be performed, and then returned to 01F0h if required, thus allowing the addition of new commands.

The extended interpreter first checks the line for ' * ' followed by a letter. It is necessary to use a symbol to stop the entry of a normal command word, though they can be used, as anything which fails the normal syntax checks can have further tests done. For example PRINT * could be used as a new command with the extended interpreter checking for token 245 and then ' * '.

As there are only three commands each with a different initial letter, only a simple test on the initial letter is being used. If any of the checks fail, then a syntax error is given, otherwise control passes to the routine for that command. The interpreter then checks the rest of the line



*MUSIC SCALE

for the required integers or variables and separators. If all is well, then the routine will be exited and allow entry of the line into the program. The syntax used for the command requires a space between the end of the command word and the first integer or variable, but the syntax checker will accept capital or small letters in the command word which allows any printout of the program to look like original Spectrum commands, except that they are preceded by a ' * '.

Software Notes

When the SHADOW ROM is paged in, routines in the BASIC ROM can be called up using RST 10H followed by the address of the routine required. In syntax time EXPT1 is used to process numeric expressions and STEND is called to mark the end of the statement. Progress is made down the line being checked by use of GETCH to find the ' * ' and NXTCH to advance past it. ERR6 points to the original address in the BASIC ROM which was intercepted to enable the new routine. CHADD is used to read the next character including spaces. During runtime EXPT1 puts the integers on the calculator stack from where they ae retrieved by FINT1 and put into the Accumulator in the case of 8 bit integers, and retrieved by FINT2 and put in the BC register pair in the case of 16 bit integers. This is a programming choice which has to be made depending on the size of the integers involved and is not done by the operating system. The runtime exit is made by a jump to END1.

The New Commands

The commands may at first sight appear to duplicate each other but that is not the case, and there is a specific purpose to each.

*PLAY enables the required channels and defines the type of sound.
*MUSIC will give a tone from the musical scale.
*SOUND will give a tone or noise of a given pitch.

*PLAY

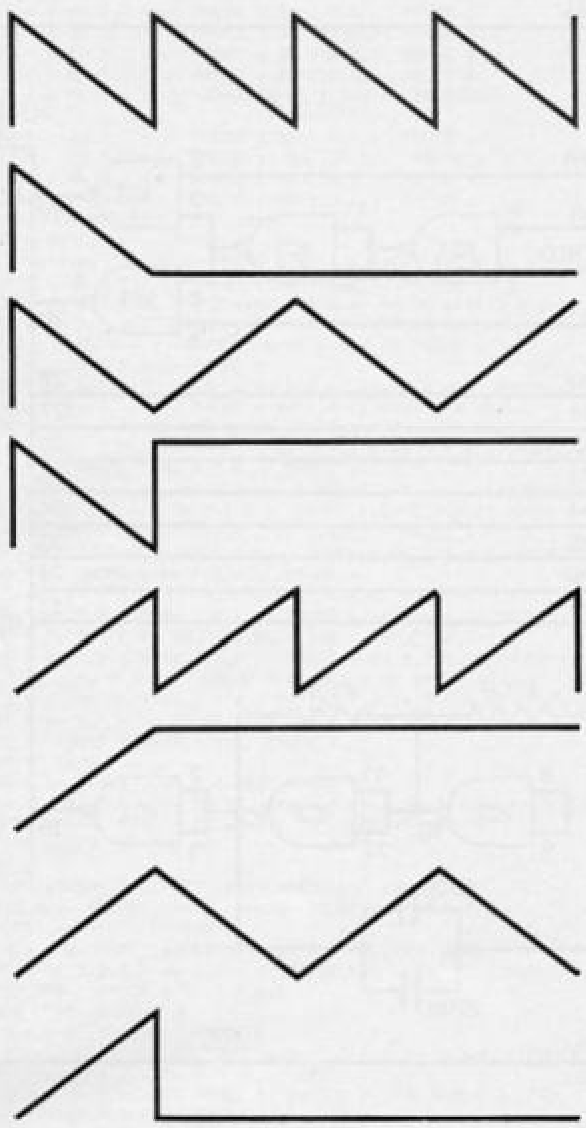
Format: tone channel, noise channel, envelope shape, envelope period.

tone channel 0-7
noise channel 0-7
envelope 0-7
duration 0-32767

This command does not actually produce any sounds but is used to enable the designated tone and noise channels selected by *MUSIC or *SOUND. It does, however, select the envelope shape and period which are common to all channels. There are three tone channels and three noise channels through which noise is output from the single noise generator. The effect of combining the channels is beyond a simple description and only experimentation can bring out the full capabilities of the generator. TONE CHANNEL selects either no channels at all, or any combination of the three where the binary bits of the number indicate the channels required. The NOISE CHANNEL selection is done in a similar manner to the tone channel selection. The

ROUTINE	*VALUE*	*FUNCTION*
GETCH	0018h	Gets character from basic into A.
NXTCH	0020h	Gets next character ignoring spaces.
CHADD	0074h	Reads next character.
ERR6	01F0h	Normal error handling address.
STEND	05B7h	Syntax exit routine.
END1	05C1h	Run-time exit routine.
EXPT1	1C82h	Syntax check for numerics — puts it on stack if alright.
FINT1	1E94h	Puts single byte integer from stack in A.
FINT2	1E99h	Puts two byte integer from shack in BC.
STAKA	2D28h	Puts A on stack.
STORE	5BF0h	Eight byte temporary store.

TABLE OF VARIABLE AND THEIR FUNCTIONS



ENVELOPE SHAPES

ENVELOPE shape is selected with reference to fig 2. and takes control of the output volume when the volume parameter on that channel is zero. If 2 or 8 is selected then an envelope of a fixed length is given. Any other choice will select a continuous sound but each has a different shape and will therefore give the sound a different timbre. The DURATION of the Envelope Period is a value between 0 > 65335.

Although the PSG could never be classed as a synthesiser the ability to modify the output waveform does allow different sounds to be produced. A sharp attack or rise time produces a percussion type sound and if followed by a rapid decay will give a sound like a snare drum. As the attack and decay are lengthened a more mellow sound will be produced. For envelope type 2, a period of around 1000 will give a sharp banjo effect and using envelope 5 with a short period of around 20 will give a machine like effect.

The duration of the note is no longer defined and will continue to sound until either the pitch is changed or it is turned off using *PLAY 0,0,0,0 which will reset all the registers to zero except register 7 which is set to 255.

```
10 REM play demonstration
20 INPUT "envelope shape";e
25 INPUT "duration";d
30 *SOUND 1,200,0
40 *PLAY 1,0,e,d
50 PRINT AT 10,2; "Press any
key to re-enter"
55 PAUSE 0
60 CLS
70 GOTO 10
```

*MUSIC

Format; channel, octave, note, volume.

channel 1-3
octave 0-6
note 1-12
volume 0-15

This command is used to generate a musical output of a tone set by OCTAVE and NOTE and output it through the designated CHANNEL at the ▶

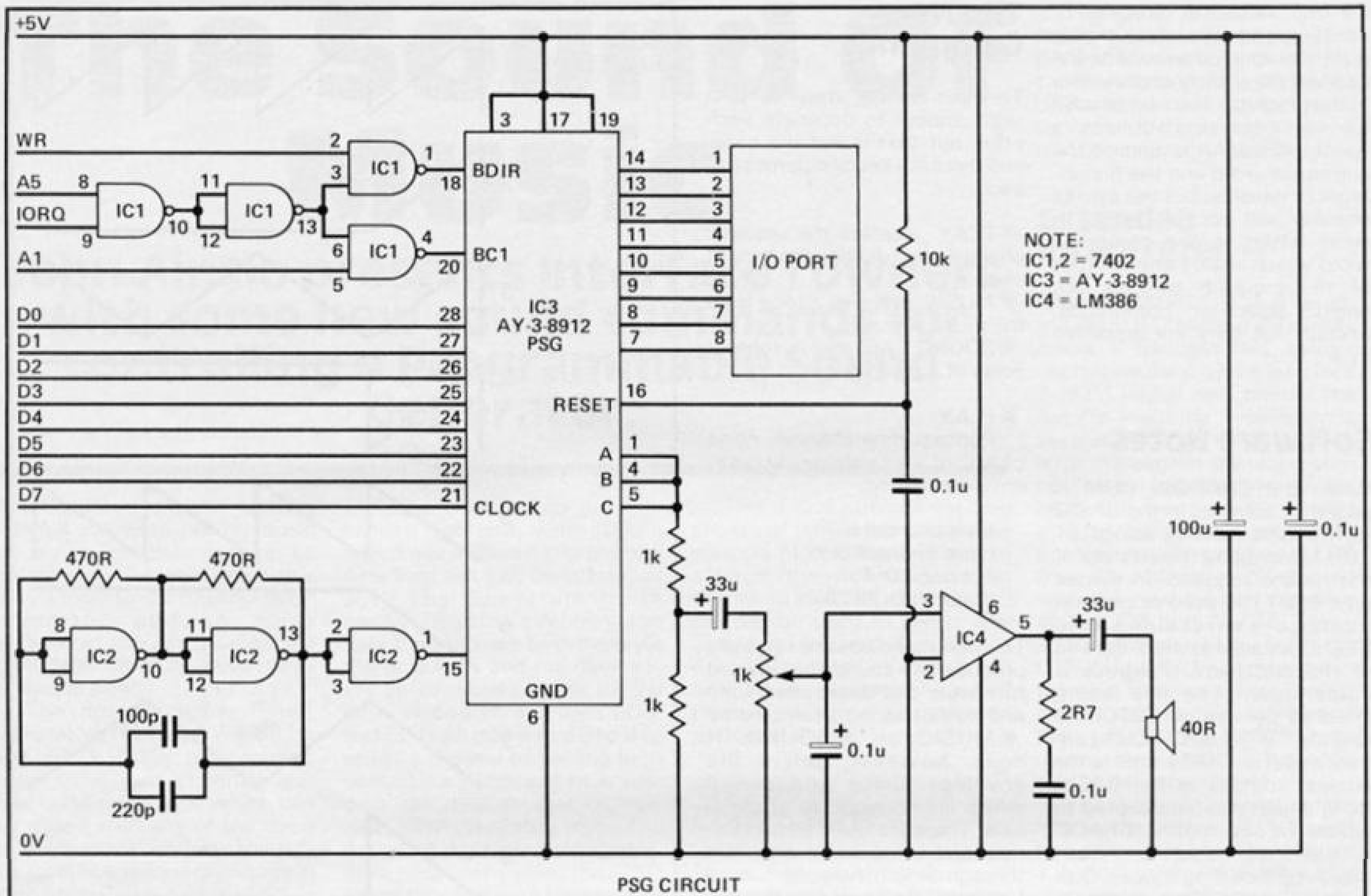


Figure 1. Diagram of the 8912 PSG Circuit.

REGISTER	FUNCTION	BIT 7	6	5	4	3	2	1	0
R0	Channel A - Tone Period	8 Bit Fine Tone A							
R1						4 Bit Course Tone A			
R2	Channel B - Tone Period	8 Bit Fine Tone B							
R3						4 Bit Course Tone B			
R4	Channel C - Tone Period	8 Bit Fine Tone C							
R5						4 Bit Course Tone C			
R6	Noise Period				5 Bit Control				
R7	Channel Enable		I/O	Noise			Tone		
				C	B	A	C	B	A
R8	Channel A - Amplitude				Env	4 Bit Control			
R9	Channel B - Amplitude				Env	4 Bit Control			
R10	Channel C - Amplitude				Env	4 Bit Control			
R11	Envelope Period	8 Bit Fine Control							
R12		8 Bit Course Control							
R13	Envelope Shape					4 Bit Control			
R14	I/O Port	8 Bit Parallel Port							

PSG REGISTERS

specified VOLUME. The command is fairly complex and, to make full use of its capabilities, experiementation is required.

There are four variables or integers involved in the command with a comma separator between each. The CHANNEL selection is pretty straightforward, selecting the channel that is required. OCTAVE 0 starts at two below middle C and increases from there. (Middle C is taken as 246.9 Hz). NOTE selects the required semi-tone within the chromatic scale and is the same as the Spectrum's BEEP command. VOLUME ranges from 0 to 15. the amplitude of the output ranges from 1 (very soft) to 15 (very loud). When the volume is set to 0, control of the output is passed to the envelope generator the parameters of which must be set in an accompanying * PLAY. * MUSIC must be used with * PLAY, which selects the channels being used and the shape of the sound envelope.

```
5 *PLAY 1,0,0,0
10 FOR o=0 TO 6
20 FOR n=1 TO 12
30 *MUSIC 1,o,n,5
40 PAUSE 20
50 NEXT n
60 NEXT o
70 *PLAY 0,0,0,0:REM turn channels off
```

***SOUND**

Format; channel, pitch, volume.

channel 1-6
pitch 0-65335
volume 0-15

*SOUND lacks the finesse of *MUSIC, since its output is not arranged in semitones and it merely blasts out whatever values are input. It can be used to generate complex sounds as well as pure tones. However, it does include a noise generator unlike *MUSIC which allows the creation of sounds which are more useful as a background to games. CHANNEL values 1-3 select a tone channel and values 4-6 select which of the three channels the single noise generator will output through. PITCH is no more than a division

```
  10 REM ***********************
  20 REM *                     *
  30 REM *      THE  STING     *
  40 REM *                     *
  50 REM *   BY SCOTT JOPLIN   *
  60 REM *                     *
  70 REM ***********************
  80 LOAD *"m";1;"psg 64000"CODE
  90 LET a=USR VAL "64000"
 100 LET a1=VAL "0": LET t=VAL ".5": LET a2=VAL "0": LET a3=VAL "0"
 110 LET s=VAL "10": REM semiquaver
 120 LET q=VAL "20": REM quaver
 130 LET a4=VAL "0": LET a5=VAL "0": LET a6=VAL "0"
 140 LET a7=VAL "0": LET a8=VAL "0": LET a9=VAL "0": LET a10=VAL "0"
 150 LET v=VAL "6": REM SET VOLUME TO SUITABLE START LEVEL
 160 REM ********** NOTE VALUES ******
 170 LET c=VAL "1": LET cs=VAL "2": LET d=VAL "3": LET ds=VAL "4"
 180 LET e=VAL "5": LET f=VAL "6": LET fs=VAL "7": LET g=VAL "8"
 190 LET gs=VAL "9": LET a=VAL "10": LET bf=VAL "11": LET b=VAL "12"
 200 *m 1,5,d,v:*m 2,4,d,v:*p 3,0,0,0: PAUSE s
 210 *m 1,5,e,v:*m 2,4,e,v: PAUSE s:*m 1,5,c,v:*m 2,4,c,v: PAUSE s
 220 *m 1,4,a,v:*m 2,3,a,v: PAUSE q:*m 1,4,b,v:*m 2,3,b,v: PAUSE s
 230 *m 1,4,g,v:*m 2,3,g,v: PAUSE q:*m 1,4,d,v:*m 2,3,d,v: PAUSE s
 240 *m 1,4,e,v:*m 2,3,e,v: PAUSE s:*m 1,4,c,v:*m 2,3,c,v: PAUSE s
 250 *m 1,3,a,v:*m 2,2,a,v: PAUSE q:*m 1,3,b,v:*m 2,2,b,v: PAUSE s
 260 *m 1,3,g,v:*m 2,2,g,v: PAUSE q:*m 1,3,d,v:*m 2,2,d,v: PAUSE s
 270 *m 1,3,e,v:*m 2,2,e,v: PAUSE s:*m 1,3,c,v:*m 2,2,c,v: PAUSE s
 280 *m 1,2,a,v:*m 2,1,a,v: PAUSE q:*m 1,2,b,v:*m 2,1,b,v: PAUSE s
 290 *m 1,2,a,v:*m 2,1,a,v: PAUSE s:*m 1,2,gs,v:*m 2,1,gs,v: PAUSE s
 300 *m 1,2,g,v:*m 2,1,g,v: PAUSE q: *p 0,0,0,0: PAUSE q: LET v=v+2
 310 *m 1,4,g,v:*m 2,3,b,v:*m 3,0,g,v: *p 7,0,0,0: PAUSE q: LET v=v-4
 320 *m 1,3,d,v:*m 2,2,b,v:*m 3,2,g,v: PAUSE s:*m 1,3,ds,v: PAUSE s
 330  IF a2=VAL "1" THEN  LET a10=VAL "1"
 340 *m 1,3,e,v:*m 2,2,c,v:*m 3,2,c,v: PAUSE s
 350 *m 1,4,c,v: PAUSE s:*m 2,3,c,v:*m 3,2,e,v: PAUSE s
 360 *m 1,3,e,v: PAUSE s:*m 1,4,c,v:*m 2,2,g,v:*m 3,1,g,v: PAUSE q
 370 *m 1,3,e,v:*m 2,3,c,v:*m 3,2,bf,v: PAUSE s
 380 *m 1,4,c,v: PAUSE s:*m 2,2,f,v:*m 3,1,f,v: PAUSE q
 390 *m 2,3,c,v:*m 3,2,a,v: PAUSE q:*m 2,2,e,v:*m 3,1,e,v: PAUSE s
 400  IF a3=VAL "1" THEN  RETURN
 410 *m 1,5,c,v:*m 2,4,e,v: PAUSE s: LET v=v+1
 420 *m 1,5,d,v:*m 2,4,f,v:*m 3,2,g,v: PAUSE s: LET v=v+1
 430 *m 1,5,ds,v:*m 2,4,fs,v: PAUSE s: LET v=v+1
 440 *m 1,5,e,v:*m 2,4,g,v:*m 3,1,g,v: PAUSE s
 450 *m 1,5,c,v:*m 2,4,e,v: PAUSE s:*m 1,5,d,v:*m 2,4,f,v:*m 3,2,g,v: PAUSE s
 460 *m 1,5,e,v:*m 2,4,g,v: PAUSE s:*m 3,1,g,v: PAUSE s
 470 *m 1,4,b,v:*m 2,4,d,v: PAUSE s:*m 1,5,d,v:*m 2,4,f,v:*m 3,2,b,v: PAUSE q
 480 *m 1,5,c,v:*m 2,4,e,v:*m 3,2,c,v: PAUSE q
 490  IF a3=VAL "2" THEN  RETURN
 500 *m 3,1,g,v: PAUSE q:*m 3,2,c,v: PAUSE q: LET v=v-2
 510 *m 1,3,d,v:*m 2,2,b,v:*m 3,2,g,v: PAUSE s:*m 1,3,ds,v: PAUSE s
 520 LET a3=VAL "1": LET v=v-1: GO SUB 340
 530 PAUSE s
 540  LET v=v+1:*m 1,4,a,v:*m 2,4,c,v:*m 3,2,ds,v: PAUSE s: LET v=v+1
 550 *m 1,4,g,v:*m 2,3,g,v: PAUSE s: LET v=v+1
 560 *m 1,4,fs,v:*m 2,3,fs,v:*m 3,1,d,v: PAUSE s
 570 *m 1,4,a,v: *m 2,3,a,v: PAUSE s:*m 1,5,c,v:*m 2,4,c,v:*m 3,2,fs,v: PAUSE s
 580 *m 1,5,e,v:*m 2,4,e,v: PAUSE s :*m 3,2,d,v: PAUSE s
 590 *m 1,5,d,v:*m 2,4,d,v: PAUSE s:*m 1,5,c,v:*m 2,4,c,v:*m 3,2,a,v: PAUSE s
 600 *m 1,4,a,v:*m 2,3,a,v: PAUSE s
 610 *m 1,5,d,v:*m 2,4,f,v:*m 3,2,b,v: PAUSE q
 620 *m 3,1,g,v: PAUSE q:*m 3,1,a,v: PAUSE q: LET v=v-1:*m 1,3,d,v
 630 *m 2,2,b,v:*m 3,1,b,v: PAUSE s: LET v=v-1:*m 1,3,ds,v: PAUSE s
 640  LET a3=VAL "2": GO SUB 340
 650 *m 3,2,g,v: PAUSE q:*m 3,3,c,v: PAUSE q:*p 3,0,0,0
 660 *m 1,5,c,v:*m 2,4,c,v: PAUSE s:*m 1,5,d,v:*m 2,4,d,v: PAUSE s
 670 *m 3,2,c,v:*p 7,0,0,0:*m 1,3,e,v:*m 2,4,e,v: PAUSE s
 680 *m 1,5,c,v:*m 2,4,c,v: PAUSE s:*m 1,5,d,v:*m 2,4,d,v:*m 3,3,c,v: PAUSE s
 690 *m 1,5,e,v:*m 2,4,e,v: PAUSE s:*m 3,1,bf,v: PAUSE s
 700 *m 1,5,c,v:*m 2,4,c,v: PAUSE s:*m 1,5,d,v:*m 2,4,d,v:*m 3,3,e,v: PAUSE s
 710 *m 1,5,c,v:*m 2,4,c,v: PAUSE s
 720 *m 1,5,e,v:*m 2,4,e,v:*m 3,1,a,v: PAUSE s:*m 1,5,c,v:*m 2,4,c,v: PAUSE s
 730 *m 1,5,d,v:*m 2,4,d,v:*m 3,3,f,v: PAUSE s:*m 1,5,e,v:*m 2,4,e,v: PAUSE s
 740 *m 3,1,gs,v: PAUSE s:*m 1,5,c,v:*m 2,4,c,v: PAUSE s
 750 *m 1,5,d,v:*m 2,4,d,v:*m 3,3,f,v: PAUSE s
 760 *m 1,5,c,v:*m 2,4,c,v: PAUSE s:*m 1,5,e,v:*m 2,4,g,v:*m 3,1,g,v: PAUSE s
 770 *m 1,5,c,v:*m 2,4,e,v: PAUSE s:*m 1,5,d,v:*m 2,4,f,v:*m 3,3,c,v: PAUSE s
 780 *m 1,5,e,v:*m 2,4,g,v: PAUSE s
 790 *m 3,1,g,v: PAUSE s:*m 1,4,b,v:*m 2,4,d,v: PAUSE s
 800 *m 1,5,d,v:*m 2,4,f,v:*m 3,2,b,v: PAUSE q
 810 IF a6=1 THEN  GO TO 970
 820 *m 1,5,c,v:*m 2,4,e,v:*m 3,2,c,v: PAUSE q
 830 *m 3,1,g,v: PAUSE q
 840  LET v=v-1:*m 3,1,a,v: PAUSE q
 850  LET v=v-1:*m 1,3,d,v:*m 2,2,b,v:*m 3,1,b,v: PAUSE s
 860 *m 1,3,ds,v: PAUSE s: LET a3=VAL "0": LET a6=1: LET v=v-2: GO TO 340
 870 LET a3=VAL "0"
 880 IF a10=1 THEN  GO TO 1510
 890 *m 1,5,c,v:*m 2,4,e,v:*m 3,2,c,v: PAUSE q:*m 3,1,g,v: PAUSE q
 900 *m 3,1,c,v+1: PAUSE s: *m 1,4,e,v:*m 2,4,c,v: PAUSE s:*p 3,0,0,0
 910 *m 1,4,f,v:*m 2,4,d,v: PAUSE s:*m 1,4,fs,v:*m 2,4,ds,v: PAUSE s
 920 *p 7,0,0,0
 930 *m 1,4,g,v:*m 2,4,e,v:*m 3,1,c,v:*m 3,1,c,v: PAUSE q
 940 *m 1,4,a,v:*m 2,4,e,v:*m 3,2,c,v: PAUSE s:*m 1,4,g,v:*m 2,4,e,v
 950  IF a9=VAL "1" THEN  GO TO 1030
 960 IF a1=VAL "1" AND a3=VAL "1" THEN  GO TO 980
 970 IF a3=VAL "1" THEN  RETURN
 980 PAUSE s
 990 *m 3,1,g,v: PAUSE s:*m 1,4,e,v:*m 2,4,c,v: PAUSE s
1000 *m 1,4,f,v:*m 2,4,d,v:*m 3,3,c,v: PAUSE s:*m 1,4,fs,v:*m 2,4,ds,v: PAUSE s
1010  IF a3=1 AND a1=1 THEN  LET a9=1: GO TO 930
1020 LET a3=VAL "1": GO SUB 930
1030 PAUSE s
1040 IF a4=VAL "1" THEN  RETURN
1050 *m 3,1,g,v: PAUSE s:*m 1,4,e,v: PAUSE s
1060 *m 1,4,c,v:*m 2,3,e,v:*m 3,2,g,v: PAUSE s:*m 1,3,g,v: PAUSE s
1070 *m 1,3,a,v:*m 2,1,f,v:*m 3,1,f,v: PAUSE s:*m 1,3,b,v: PAUSE s
1080 *m 1,4,c,v:*m 2,3,f,v:*m 3,2,a,v: PAUSE s:*m 1,4,d,v: PAUSE s
1090 *m 1,4,e,v:*m 2,2,f,v:*m 3,2,f,v: PAUSE s:*m 1,4,d,v: PAUSE s
1100 *m 1,4,c,v:*m 2,3,f,v:*m 3,2,gs,v: PAUSE s:*m 1,4,d,v: PAUSE s
1110  IF a1=1 THEN  RETURN
1120 *m 1,3,g,v:*m 2,2,e,v:*m 3,2,e,v: PAUSE s:*m 1,4,e,v: PAUSE s
1130 *m 1,4,f,v:*m 2,3,e,v:*m 3,2,g,v: PAUSE s: *m 1,4,g,v: PAUSE s
1140 *m 1,4,a,v:*m 2,1,g,v:*m 3,1,g,v: PAUSE s:*m 1,4,g,v: PAUSE s
1150 *m 1,4,e,v:*m 2,3,e,v:*m 3,2,g,v: PAUSE s:*m 1,4,f,v: PAUSE s
1160 LET a3=0:  LET a4=1: GO SUB 930:  LET a4=0
1170 *m 3,2,e,v: PAUSE s:*m 1,4,g,v: PAUSE s
1180 *m 1,4,a,v:*m 2,2,ds,v:*m 3,2,ds,v: PAUSE s:*m 1,4,bf,v: PAUSE s
1190 *m 1,4,b,v:*m 2,4,d,v:*m 3,2,d,v: PAUSE s:*p 4,0,0,0: PAUSE t
1200 *p 7,0,0,0: PAUSE s:*m 3,2,g,v: PAUSE s:*p 6,0,0,0: PAUSE t:*p 7,0,0,0
1210 *m 2,4,c,v: PAUSE s:*m 3,2,d,v: PAUSE s:*m 1,4,a,v: PAUSE s
1220 *m 1,4,fs,v:*m 2,3,c,v:*m 3,2,a,v: PAUSE s:*m 1,4,d,v: PAUSE s
1230 *m 1,4,g,v:*m 2,3,b,v:*m 3,2,g,v: PAUSE q
1240 *m 3,1,f,v: PAUSE q:*m 3,1,e,v: PAUSE s:*m 1,4,e,v:*m 2,4,c,v: PAUSE s
1250 *m 1,4,f,v:*m 2,4,d,v:*m 3,1,d,v: PAUSE s:*m 1,4,fs,v:*m 2,4,ds,v
1260  PAUSE s: LET a4=0: LET a1=1: GO SUB 930
1270 *m 1,4,c,v:*m 2,2,e,v:*m 3,2,e,v: PAUSE q:*m 2,3,e,v:*m 3,2,g,v: PAUSE q
1280 *m 3,2,c,v:*m 2,2,c,v: PAUSE s:*m 1,3,g,v: PAUSE s
1290 LET v=v-1:*m 1,3,fs,v:*m 2,3,e,v:*m 3,2,bf,v: PAUSE s:*m 1,3,g,v: PAUSE s
1300 LET v=v-1:*m 1,4,c,v:*m 2,3,a,v:*m 3,2,f,v: PAUSE q:*p 3,0,0,0: PAUSE t
1310 *p 7,0,0,0:*m 1,3,a,v:*m 2,3,f,v: PAUSE s:*m 1,4,c,v: PAUSE s
1320 *m 2,3,ds,v:*m 3,2,fs,v: PAUSE s:*m 1,3,a,v: PAUSE s:*p 1,0,0,0: PAUSE t
1330 *p 7,0,0,0:*m 1,4,c,v: PAUSE s:*m 1,3,a,v: PAUSE s
1340 *m 1,3,g,v:*m 2,3,e,v:*m 3,2,g,v: PAUSE s:*m 1,4,c,v: PAUSE s
1350 *p 1,0,0,0: PAUSE t:*p 7,0,0,0:*m 1,4,e,v: PAUSE s:*m 1,4,g,v: PAUSE s
1360 *p 1,0,0,0: PAUSE t:*p 7,0,0,0: PAUSE s:*m 1,4,e,v: PAUSE s
1370 *p 1,0,0,0: PAUSE t:*p 7,0,0,0:*m 1,4,c,v: PAUSE s:*m 1,3,g,v: PAUSE s
1380 *m 1,3,a,v:*m 2,3,fs,v:*m 3,2,d,v: PAUSE q:*p 1,0,0,0: PAUSE t
1390 *p 7,0,0,0:*m 1,4,c,v: PAUSE q:*m 1,4,e,v:*m 2,3,f,v:*m 3,1,g,v: PAUSE s
1400 *m 1,4,d,v: PAUSE s:*m 3,1,b,v: PAUSE s:*m 1,4,c,v:*m 2,3,e,v
1410  PAUSE s:*m 3,2,c,v: PAUSE q:*m 3,1,g,v: PAUSE q
1420  IF a2=1 THEN  GO TO 1470
1430 *m 3,1,e,v: PAUSE s:*m 1,4,e,v:*m 2,4,c,v: PAUSE s
1440 *m 1,4,f,v:*m 2,4,d,v:*m 3,1,d,v: PAUSE s:*m 1,4,fs,v:*m 2,4,ds,v: PAUSE s
1450 LET a4=VAL "0": LET a1=VAL "0":  LET a2=VAL "1": LET a9=VAL "0"
1460 LET a3=VAL "0": LET v=v+2: GO SUB 930
1470 *m 3,1,c,v: PAUSE q
1480 LET a1=VAL "0": LET a3=VAL "0": LET a4=VAL "0": LET a9=VAL "0"
1490 *p 1,0,0,0: *m 1,3,d,v: PAUSE s:*m 1,3,ds,v: PAUSE s:*p 7,0,0,0
1500  IF a10=0 THEN  LET v=v-1: GO TO 330
1510 *m 1,5,c,v:*m 2,4,e,v:*m 3,2,c,v: PAUSE q:*m 3,1,g,v: PAUSE q
1520 *p 4,0,0,0: PAUSE t:*p 7,0,0,0:*m 3,1,c,v: PAUSE q
1530 OUT 221,7: OUT 223,255:  PAUSE q:*p 7,0,0,0
1540 IF a7=VAL "1" THEN  LET a5=VAL "1"
1550 *m 1,3,f,v:*m 2,3,d,v:*m 3,1,f,v: PAUSE q
1560 *m 1,3,e,v:*m 2,3,cs,v:*m 3,2,a,v: PAUSE s:*m 1,3,f,v:*m 2,3,d,v: PAUSE s
1570 *m 3,1,a,v: PAUSE s:*m 1,3,e,v:*m 2,3,cs,v: PAUSE s
1580 *m 1,3,f,v:*m 2,3,d,v:*m 3,2,a,v: PAUSE q: *m 3,1,f,v:*p 4,0,0,0: PAUSE s
1590 *m 1,3,a,v:*m 2,3,f,v:*p 5,0,0,0: PAUSE s
1600 *p 7,0,0,0:*m 1,4,d,v:*m 3,2,a,v: PAUSE s:*m 1,3,a,v: PAUSE s
1610 *m 1,4,c,v:*m 2,1,a,v:*m 3,1,a,v: PAUSE s:*m 1,4,d,v: PAUSE s
1620 *m 1,4,c,v:*m 2,2,a,v:*m 3,2,f,v: PAUSE s:*m 1,3,a,v: PAUSE s
1630 *m 1,3,g,v:*m 2,3,e,v:*m 3,2,c,v: PAUSE q
1640 *m 1,3,fs,v:*m 2,3,ds,v:*m 3,3,c,v: PAUSE s:*m 1,3,g,v
1650 *m 2,3,e,v: PAUSE s:*m 3,1,g,v: PAUSE s:*m 1,3,fs,v:*m 2,3,ds,v: PAUSE s
1660 *m 1,3,g,v:*m 2,3,e,v:*m 3,3,e,v: PAUSE q
1670 *m 1,2,c,v:*m 2,2,c,v:*m 3,2,c,v: PAUSE s:*m 1,4,c,v: PAUSE s
1680 *m 1,4,f,v:*m 2,3,g,v:*m 3,3,c,v: PAUSE s:*m 1,4,c,v: PAUSE s
1690 *m 1,4,d,v:*m 2,1,g,v:*m 3,1,g,v: PAUSE s:*m 1,4,e,v: PAUSE s
1700 *m 1,4,d,v:*m 2,3,c,v:*m 3,2,e,v: PAUSE s:*m 1,4,c,v: PAUSE s
1710  IF a5=VAL "1" THEN  GO TO 1890
1720 *m 1,4,d,v:*m 2,3,b,v:*m 3,1,g,v: PAUSE q
1730 *m 1,4,cs,v:*m 2,3,bf,v:*m 3,2,g,v: PAUSE s:*m 1,4,d,v:*m 2,3,b,v: PAUSE s
1740 *m 3,1,b,v: PAUSE s:*m 1,4,cs,v:*m 2,3,bf,v: PAUSE s
1750 *m 1,4,d,v:*m 2,3,b,v: PAUSE q
1760 *m 1,1,g,v:*m 2,1,g,v:*m 3,1,g,v: PAUSE s:*m 1,4,f,v: PAUSE s
1770 *m 1,4,a,v:*m 2,3,b,v:*m 3,2,g,v: PAUSE s:*m 1,4,f,v: PAUSE s
1780 *m 1,4,g,v:*m 2,2,d,v:*m 3,2,d,v: PAUSE s:*m 1,4,a,v: PAUSE s
1790 *m 1,4,g,v:*m 2,3,g,v:*m 3,2,g,v: PAUSE s:*m 1,4,f,v: PAUSE s
1800 LET v=v+2: *m 1,5,c,v:*m 2,3,fs,v: PAUSE s
1810 *p 6,0,0,0: PAUSE t:*p 7,0,0,0: PAUSE s
1820 OUT 221,7: OUT 223,255: PAUSE t:*p 7,0,0,0: PAUSE 3*s
1830 OUT 221,7: OUT 223,255: PAUSE s
1840 *m 1,4,a,v:*m 2,3,fs,v:*m 3,2,g,v:*p 7,0,0,0: PAUSE q
1850 *m 1,4,g,v:*m 2,3,e,v:*m 3,2,e,v: PAUSE q
1860  LET v=v-3:*p 1,0,0,0: *m 1,3,g,v: PAUSE s:*m 1,3,e,v: PAUSE s
1870 *m 1,3,g,v: PAUSE q:*m 1,3,e,v: PAUSE q:*m 2,3,d,v:*m 3,1,f,v
1880  LET a7=VAL "1": GO TO 1540
1890 *m 1,3,a,v:*m 2,2,f,v:*m 3,1,f,v: PAUSE s:*m 1,3,gs,v: PAUSE s
1900 *m 1,3,a,v:*m 2,2,d,v:*m 3,1,d,v: PAUSE s:*m 1,4,g,v:*m 2,3,a,v: PAUSE s
1910 *m 3,1,e,v: PAUSE s:*m 1,4,f,v: PAUSE s:*m 3,1,f,v: PAUSE s
1920 *m 1,4,c,v: PAUSE s:*m 1,4,e,v:*m 2,3,g,v:*m 3,1,g,v: PAUSE s
1930 *m 1,4,ds,v: PAUSE s:*m 1,4,e,v:*m 2,3,e,v:*m 3,2,g,v: PAUSE s
1940 *m 1,4,a,v: PAUSE s:*m 2,3,ds,v:*m 3,2,fs,v: PAUSE s
1950 *m 1,5,c,v: PAUSE s:*m 1,4,g,v:*m 2,3,e,v:*m 3,2,g,v: PAUSE s
1960 *m 1,4,e,v: PAUSE s:*m 1,4,c,v:*m 2,3,fs,v:*m 3,1,a,v: PAUSE q
1970 *p 4,0,0,0: PAUSE t:*m 3,1,d,v:*p 7,0,0,0: PAUSE q
1980 *m 1,4,e,v:*m 2,3,f,v:*m 3,1,g,v: PAUSE s:*m 1,4,d,v: PAUSE s
1990 *m 3,1,b,v: PAUSE s:*m 1,4,c,v:*m 2,3,e,v: PAUSE s:*m 3,2,c,v: PAUSE q
2000  IF a8=0 THEN  GO TO 2020
2010 GO TO 2050
2020 *p 1,0,0,0:*m 1,3,g,v: PAUSE s:*m 1,3,e,v: PAUSE s:*m 1,3,g,v: PAUSE q
2030 *m 1,3,e,v: PAUSE q: LET a8=VAL "1": LET a7=VAL "0": LET a5=VAL "0"
2040 *p 7,0,0,0: GO TO 1540
2050 LET v=v+1
2060 *m 3,1,g,v: PAUSE q: LET v=v+1:*m 1,5,c,v:*m 2,4,e,v:*m 3,1,c,v: PAUSE q
2070 *p 0,0,0,0: STOP
2080 ERASE "m";1;"THE STING"
2090 SAVE *"m";1;"THE STING" LINE 10
2100 VERIFY *"m";1;"THE STING"
```

HEX DISASSEMBLY OF *COMMANDS

```
FA00 01 08 FA ED
FA04 43 B7 5C C9
FA08 D7 18 00 FE
FA0C 2A C2 F0 01
FA10 D7 20 00 CB
FA14 EF FE 6D CA
FA18 3F FA FE 73
FA1C CA EF FA FE
FA20 70 CA 83 FB
FA24 C3 F0 01 F1
FA28 06 8E 06 30
FA2C 06 D7 05 83
FA30 05 34 05 E9
FA34 04 A2 04 60
FA38 04 20 04 E6
FA3C 03 AE 03 D7
FA40 74 00 FE 20
FA44 28 08 FE 3A
FA48 28 04 FE 0D
FA4C 20 F1 D7 82
FA50 1C FE 2C C2
FA54 F0 01 D7 20
FA58 00 D7 82 1C
FA5C FE 2C C2 F0
FA60 01 D7 20 00
FA64 D7 82 1C FE
FA68 2C D7 20 00
FA6C D7 82 1C CD
FA70 B7 05 D7 94
FA74 1E FE 10 30
FA78 73 A7 20 02
FA7C C6 10 32 F1
FA80 5B D7 94 1E
FA84 FE 0D 30 64
FA88 A7 28 61 D6

FA8C 01 87 21 27
FA90 FA 06 00 4F
FA94 09 7E 23 46
FA98 4F ED 43 F3
FA9C 5B D7 94 1E
FAA0 FE 07 30 48
FAA4 3C ED 4B F4
FAA8 5B 47 3A F3
FAAC 5B 18 07 CB
FAB0 19 CB B9 CB
FAB4 1F A7 10 F7
FAB8 32 F3 5B 79
FABC 32 F5 5B D7
FAC0 94 1E FE 04
FAC4 30 26 A7 28
FAC8 23 F5 C6 07
FACC 32 F0 5B F1
FAD0 D6 01 87 32
FAD4 F2 5B 3C 32
FAD8 F4 5B 06 03
FADC 21 F0 5B 7E
FAE0 D3 DD 23 7E
FAE4 D3 DF 23 10
FAE8 F6 C3 C1 05
FAEC D7 F9 24 D7
FAF0 74 00 FE 20
FAF4 28 08 FE 3A
FAF8 28 04 FE 0D
FAFC 20 F1 D7 82
FB00 1C FE 2C C2
FB04 F0 01 D7 20
FB08 00 D7 82 1C
FB0C FE 2C C2 F0
FB10 01 D7 20 00
FB14 D7 82 1C CD

FB18 B7 05 D7 94
FB1C 1E FE 10 30
FB20 35 A7 20 02
FB24 C6 10 32 F1
FB28 5B D7 99 1E
FB2C 79 32 F3 5B
FB30 78 32 F5 5B
FB34 D7 94 1E FE
FB38 07 30 1B A7
FB3C 28 18 F5 F5
FB40 FE 04 30 15
FB44 C6 07 32 F0
FB48 5B F1 D6 01
FB4C 87 32 F2 5B
FB50 3C 32 F4 5B
FB54 18 1B D7 F9
FB58 24 C6 04 32
FB5C F0 5B F1 3E
FB60 06 32 F2 5B
FB64 32 F4 5B 3A
FB68 F3 5B 32 F5
FB6C 5B FE 20 30
FB70 E5 06 03 21
FB74 F0 5B 7E D3
FB78 DD 23 7E D3
FB7C DF 23 10 F6
FB80 C3 C1 05 D7
FB84 74 00 FE 20
FB88 28 08 FE 3A
FB8C 28 04 FE 0D
FB90 20 F1 D7 82
FB94 1C FE 2C C2
FB98 F0 01 D7 20
FB9C 00 D7 82 1C
FBA0 FE 2C C2 F0
FBA4 01 D7 20 00
FBA8 D7 82 1C FE

FBAC 2C C2 F0 01
FBB0 D7 20 00 D7
FBB4 82 1C CD B7
FBB8 05 D7 99 1E
FBBC 79 32 F1 5B
FBC0 78 32 F3 5B
FBC4 3E 0B 32 F0
FBC8 5B 3C 32 F2
FBCC 5B D7 94 1E
FBD0 FE 09 30 47
FBD4 A7 28 02 C6
FBD8 07 32 F5 5B
FBDC 3E 0D 32 F4
FBE0 5B 3E 07 32
FBE4 F6 5B D7 94
FBE8 1E FE 08 30
FBEC 2E CB 27 CB
FBF0 27 CB 27 32
FBF4 F7 5B D7 94
FBF8 1E FE 08 30
FBFC 1E 21 F7 5B
FC00 B6 2F 32 F7
FC04 5B FE FF 28
FC08 15 06 04 21
FC0C F0 5B 7E D3
FC10 DD 23 7E D3
FC14 DF 23 10 F6
FC18 C3 C1 05 D7
FC1C F9 24 AF 32
FC20 F0 5B 32 F1
FC24 5B 06 0E 21
FC28 F0 5B 7E D3
FC2C DD 34 23 7E
FC30 D3 DF 10 F3
FC34 3E 07 D3 DD
FC38 3E FF D3 DF
FC3C C3 C1 05 00
```

of the clock frequency. A small number will probably have your dog jumping around as the sound is ultrasonic. VOLUME is the same as the other commands, 1 (quiet) to 15 (very loud) and 0 to enable the envelope to control the output.

*SOUND must be enabled with the appropriate selection in the *PLAY to enable the selected channel and is turned off by *PLAY 0,0,0,0.

```
10 REM sound demo
15 *PLAY 1,0,0,0
20 FOR p=1 TO 500 STEP 10
30 *SOUND 1,8 *p,5
40 PAUSE 10
50 *SOUND 1,4 *p,5
60 PAUSE 10
65 *SOUND 1,12 *p,5
67 PAUSE 10
70 NEXT p
80 *PLAY 0,0,0,0
```

The Shape of Things To Come

It is not possible to tune a keyed instrument to a diatonic scale exactly, and so the equal tempered scale was evolved, in which every interval has the same frequency ratio of 12 2 (1.05946) — a complete octave having a ratio of 2:1, being effected in twelve notes.

The tones generated by the PSG are obtained by dividing the clock frequency input by 16 and then programmable divisions of this frequency by PITCH or preset divisions for NOTE and OCTAVE. The PSG has a maximum clock frequency of 2MHz and values as close to this as possible are preferred. The table within the program uses as its base the C two octaves below middle C — a frequency of 65Hz which is the base for all calculations and is OCTAVE 0, NOTE 1. The components specified in the circuit gave a frequency of approximately 1.86MHz. There is a fixed division of 16 inside the PSG and this resulting frequency of 116.25KHz is the output when both course and fine registers of a channel are set to zero. This frequency is divided by 1777 to give 65Hz, so 1777 is the base value required for the table contained in the program. This base value can be altered to tune the PSG to a piano or other musical instrument and the other 11 values can be calculated by repeated division to obtain the other values and inserted into the program.

C	1777	(65.4Hz)
C#	1678	(69.3Hz)
D	1584	(73.4Hz)
D#	1495	(77.8Hz)
E	1411	(82.4Hz)
F	1332	(87.3Hz)
F#	1257	(92.5Hz)
G	1186	(98.0Hz)
Ab	1120	(103.8Hz)
A	1056	(110.0Hz)
Bb	998	(116.6Hz)
B	942	(122.5Hz)

```
                0010 ;              "SOURCE CODE AND ASSEMBLY LANGUAGE"
                0020 ;              "FOR EXTENDED BASIC * COMMANDS"
                0030 ;              "Io HARDWARE AND SOFTWARE"
                0040 ;              "COPYRIGHT 1985"
                0050 ;
                0060 ;
FA00            0070        ORG  64000
                0080 ;
                0090 ;
0018            0100 GETCH  EQU  18H
0020            0110 NXTCH  EQU  20H
0074            0120 CHADD  EQU  74H
01F0            0130 ERR6   EQU  01F0H
05B7            0140 STEND  EQU  05B7H
05C1            0150 END1   EQU  05C1H
1C82            0160 EXPT1  EQU  01C82H
1E94            0170 FINT1  EQU  01E94H
1E99            0180 FINT2  EQU  1E99H
2D28            0190 STAKA  EQU  2D28H
5BF0            0200 MSTR   EQU  5BF0H
                0210 ;
                0220 ;
FA00 0108FA     0230 INIT   LD   BC,START
FA03 ED4287FC   0240        LD   (23735),BC
FA07 C9         0250        RET
                0260 ;
                0270 ;
FA08 D7         0280 START  RST  10H
FA09 1800       0290        DEFW GETCH
FA0B FE2A       0300        CP   "*"
FA0D C2F001     0310        JP   NZ,ERR6
FA10 D7         0320        RST  10H
FA11 2000       0330        DEFW NXTCH
FA13 CBEF       0340        SET  5,A
FA15 FE6D       0350        CP   "m"
FA17 CA3FFF     0360        JP   Z,MUSIC
FA1A FE73       0370        CP   "s"
FA1C CAEFFA     0380        JP   Z,SOUND
FA1F FE70       0390        CP   "p"
FA21 CA83FB     0400        JP   Z,PLAY
FA24 C3F001     0410        JP   ERR6
                0420 ;
                0430 ;
                0440 ;
                0450 ;
FA27 F106       0460 TABLE DEFW 1777
FA29 3E06       0470        DEFW 1678
FA2B 3006       0480        DEFW 1584
FA2D D705       0490        DEFW 1495
FA2F 8305       0500        DEFW 1411
FA31 3405       0510        DEFW 1332
FA33 E904       0520        DEFW 1257
FA35 A204       0530        DEFW 1196
FA37 6004       0540        DEFW 1120
FA39 2004       0550        DEFW 1056
FA3B E603       0560        DEFW 998
FA3D AE03       0570        DEFW 942
                0580 ;
                0590 ;
                0600 ;              "MUSIC  1-3"
                0610 ;              "OCTAVE 0-6"
                0620 ;              "NOTE   1-12"
                0630 ;              "VOLUME 0-15"
                0640 ;
                0650 ;
                0660 ;              "MUSIC SYNTAX"
                0670 ;
                0680 ;
FA3F D7         0690 MUSIC RST  10H
FA40 7400       0700        DEFW CHADD
FA42 FE20       0710        CP   " "
FA44 2808       0720        JR   Z,M1
FA46 FE3A       0730        CP   ":"
FA48 2804       0740        JR   Z,M1
FA4A FE0D       0750        CP   0DH
FA4C 20F1       0760        JR   NZ,MUSIC
FA4E D7         0770 M1     RST  10H
FA4F 821C       0780        DEFW EXPT1
FA51 FE2C       0790        CP   ","
FA53 C2F001     0800        JP   NZ,ERR6
FA56 D7         0810        RST  10H
FA57 2000       0820        DEFW NXTCH
FA59 D7         0830        RST  10H
FA5A 821C       0840        DEFW EXPT1
FA5C FE2C       0850        CP   ","
FA5E C2F001     0860        JP   NZ,ERR6
FA61 D7         0870        RST  10H
FA62 2000       0880        DEFW NXTCH
FA64 D7         0890        RST  10H
FA65 821C       0900        DEFW EXPT1
FA67 FE2C       0910        CP   ","
FA69 D7         0920        RST  10H
FA6A 2000       0930        DEFW NXTCH
FA6C D7         0940        RST  10H
FA6D 821C       0950        DEFW EXPT1
FA6F CDB705     0960        CALL STEND
                0970 ;
                0980 ;              "MUSIC RUN-TIME"
                0990 ;
FA72 D7         1000 VM     RST  10H
FA73 941E       1010        DEFW FINT1
FA75 FE10       1020        CP   16
FA77 3073       1030        JR   NC,ERRM
FA79 A7         1040        AND  A
FA7A 2002       1050        JR   NZ,VM1
FA7C C610       1060        ADD  16
FA7E 32F15B     1070 VM1    LD   (MSTR+1),A
                1080 ;
                1090 ;
FA81 D7         1100 NOTE   RST  10H
FA82 941E       1110        DEFW FINT1
FA84 FE0D       1120        CP   13
FA86 3064       1130        JR   NC,ERRM
FA88 A7         1140        AND  A
FA89 2861       1150        JR   Z,ERRM
FA8B D601       1160        SUB  1
FA8D 87         1170        ADD  A
FA8E 2127FA     1180        LD   HL,TABLE
FA91 0600       1190        LD   B,0
FA93 4F         1200        LD   C,A
FA94 09         1210        ADD  HL,BC
FA95 7E         1220        LD   A,(HL)
FA96 23         1230        INC  HL
FA97 46         1240        LD   B,(HL)
FA98 4F         1250        LD   C,A
FA99 ED43F35B   1260        LD   (MSTR+3),BC
                1270 ;
                1280 ;
FA9D D7         1290 OCT    RST  10H
FA9E 941E       1300        DEFW FINT1
FAA0 FE07       1310        CP   7
FAA2 3048       1320        JR   NC,ERRM
FAA4 3C         1330        INC  A
FAA5 ED4BF45B   1340        LD   BC,(MSTR+4)
FAA9 47         1350        LD   B,A
FAAA 3AF35B     1360        LD   A,(MSTR+3)
FAAD 1807       1370        JR   COUNT
FAAF CB19       1380 HALF   RR   C
FAB1 CBB9       1390        RES  7,C
FAB3 CB1F       1400        RR   A
FAB5 A7         1410        AND  A
FAB6 10F7       1420 COUNT  DJNZ HALF
FAB8 32F35B     1430        LD   (MSTR+3),A
FABB 79         1440        LD   A,C
FABC 32F55B     1450        LD   (MSTR+5),A
                1460 ;
                1470 ;
FABF D7         1480 CHAN   RST  10H
FAC0 941E       1490        DEFW FINT1
FAC2 FE04       1500        CP   4
FAC4 3026       1510        JR   NC,ERRM
FAC6 A7         1520        AND  A
FAC7 2823       1530        JR   Z,ERRM
FAC9 F5         1540        PUSH AF
FACA C607       1550        ADD  7
FACC 32F05B     1560        LD   (MSTR),A
FACF F1         1570        POP  AF
FAD0 D601       1580        SUB  1
FAD2 87         1590        ADD  A
FAD3 32F25B     1600        LD   (MSTR+2),A
FAD6 3C         1610        INC  A
FAD7 32F45B     1620        LD   (MSTR+4),A
FADA 0603       1630        LD   B,3
                1640 ;
                1650 ;
FADC 21F05B     1660 OUTPT  LD   HL,MSTR
FADF 7E         1670 LOOP   LD   A,(HL)
FAE0 D3DD       1680        OUT  (221),A
FAE2 23         1690        INC  HL
FAE3 7E         1700        LD   A,(HL)
```

```
FAE4 D3DF        OUT  (223),A
FAE6 23          INC  HL
FAE7 10F6        DJNZ LOOP
FAE9 C3C105      JP   END1
            ;
            ;
FAEC D7     ERRM RST  10H
FAED F924        DEFW 24F9H
            ;
            ;
            ;
            ;        "CHANS  1-6"
            ;        "PERIOD 0-65535"
            ;        "VOL    0-15"
            ;
            ;
            ;        "SOUND SYNTAX"
            ;
            ;
FAEF D7     SOUND RST 10H
FAF0 7400        DEFW CHADD
FAF2 FE20        CP   " "
FAF4 2808        JR   Z,S1
FAF6 FE3A        CP   ":"
FAF8 2804        JR   Z,S1
FAFA FE0D        CP   0DH
FAFC 20F1        JR   NZ,SOUND
FAFE D7     S1   RST  10H
FAFF 821C        DEFW EXPT1
FB01 FE2C        CP   ","
FB03 C2F001      JP   NZ,ERR6
FB06 D7          RST  10H
FB07 2000        DEFW NXTCH
FB09 D7          RST  10H
FB0A 821C        DEFW EXPT1
FB0C FE2C        CP   ","
FB0E C2F001      JP   NZ,ERR6
FB11 D7          RST  10H
FB12 2000        DEFW NXTCH
FB14 D7          RST  10H
FB15 821C        DEFW EXPT1
FB17 CD8705      CALL STEND
            ;
            ;
            ;        "SOUND RUN-TIME"
            ;
            ;
            ;
FB1A D7     VOLS RST  10H
FB1B 941E        DEFW FINT1
FB1D FE10        CP   16
FB1F 3035        JR   NC,ERRS
FB21 A7          AND  A
FB22 2002        JR   NZ,VS1
FB24 C610        ADD  16
FB26 32F15B VS1  LD   (MSTR+1),A
            ;
            ;
FB29 D7     PER  RST  10H
FB2A 991E        DEFW FINT2
FB2C 79          LD   A,C
FB2D 32F35B      LD   (MSTR+3),A
FB30 78          LD   A,B
FB31 32F55B      LD   (MSTR+5),A
            ;
            ;
FB34 D7     CHANS RST 10H
FB35 941E        DEFW FINT1
FB37 FE07        CP   7
FB39 301B        JR   NC,ERRS
FB3B A7          AND  A
FB3C 2818        JR   Z,ERRS
FB3E F5          PUSH AF
FB3F F5          PUSH AF
FB40 FE04        CP   4
FB42 3015        JR   NC,C1
FB44 C607        ADD  7
FB46 32F05B      LD   (MSTR),A
FB49 F1          POP  AF
FB4A D601        SUB  1
FB4C 87          ADD  A
FB4D 32F25B      LD   (MSTR+2),A
FB50 3C          INC  A
FB51 32F45B      LD   (MSTR+4),A
FB54 1818        JR   C2
            ;
            ;
FB56 D7     ERRS RST  10H
FB57 F924        DEFW 24F9H
            ;
FB59 C604   C1   ADD  4
FB5B 32F05B      LD   (MSTR),A
FB5E F1          POP  AF
FB5F 3E06        LD   A,6
FB61 32F25B      LD   (MSTR+2),A
FB64 32F45B      LD   (MSTR+4),A
FB67 3AF35B      LD   A,(MSTR+3)
FB6A 32F55B      LD   (MSTR+5),A
FB6D FE20        CP   32
FB6F 30E5        JR   NC,ERRS
FB71 0603   C2   LD   B,3
FB73 21F05B      LD   HL,MSTR
FB76 7E     LOOPS LD  A,(HL)
FB77 D3DD        OUT  (221),A
FB79 23          INC  HL
FB7A 7E          LD   A,(HL)
FB7B D3DF        OUT  (223),A
FB7D 23          INC  HL
FB7E 10F6        DJNZ LOOPS
FB80 C3C105      JP   END1
            ;
            ;
            ;        "TONE   0-7"
            ;        "NOISE  0-7"
            ;        "ENV    0-8"
            ;        "PERIOD 0-65535"
            ;
            ;        "PLAY SYNTAX"
            ;
FB83 D7     PLAY RST  10H
FB84 7400        DEFW CHADD
FB86 FE20        CP   " "
FB88 2808        JR   Z,P1
FB8A FE3A        CP   ":"
FB8C 2804        JR   Z,P1
FB8E FE0D        CP   0DH
FB90 20F1        JR   NZ,PLAY
FB92 D7     P1   RST  10H
FB93 821C        DEFW EXPT1
FB95 FE2C        CP   ","
FB97 C2F001      JP   NZ,ERR6
FB9A D7          RST  10H
FB9B 2000        DEFW NXTCH
FB9D D7          RST  10H
FB9E 821C        DEFW EXPT1
FBA0 FE2C        CP   ","
FBA2 C2F001      JP   NZ,ERR6
FBA5 D7          RST  10H
FBA6 2000        DEFW NXTCH
FBA8 D7          RST  10H
FBA9 821C        DEFW EXPT1
FBAB FE2C        CP   ","
FBAD C2F001      JP   NZ,ERR6
FBB0 D7          RST  10H
FBB1 2000        DEFW NXTCH
FBB3 D7          RST  10H
FBB4 821C        DEFW EXPT1
FBB6 CD8705      CALL STEND
            ;
            ;
            ;        "PLAY RUN-TIME"
            ;
            ;
FBB9 D7     PERPL RST 10H
FBBA 991E        DEFW FINT2
FBBC 79          LD   A,C
FBBD 32F15B      LD   (MSTR+1),A
FBC0 78          LD   A,B
FBC1 32F35B      LD   (MSTR+3),A
FBC4 3E0B        LD   A,11
FBC6 32F05B      LD   (MSTR),A
FBC9 3C          INC  A
FBCA 32F25B      LD   (MSTR+2),A
            ;
            ;
FBCD D7     ENV  RST  10H
FBCE 941E        DEFW FINT1
FBD0 FE09        CP   9
FBD2 3047        JR   NC,ERRP
FBD4 A7          AND  A
```

```
FBD5 2802          JR   Z,ENV1
FBD7 C607          ADD  7
FBD9 32F55B  ENV1  LD   (MSTR+5),A
FBDC 3E0D          LD   A,13
FBDE 32F45B        LD   (MSTR+4),A
             ;
             ;
FBE1 3E07    NOISE LD   A,7
FBE3 32F65B        LD   (MSTR+6),A
FBE6 D7            RST  10H
FBE7 941E          DEFW FINT1
FBE9 FE08          CP   8
FBEB 302E          JR   NC,ERRP
FBED CB27          SLA  A
FBEF CB27          SLA  A
FBF1 CB27          SLA  A
FBF3 32F75B        LD   (MSTR+7),A
             ;
             ;
FBF6 D7      TONE  RST  10H
FBF7 941E          DEFW FINT1
FBF9 FE08          CP   8
FBFB 301E          JR   NC,ERRP
FBFD 21F75B        LD   HL,MSTR+7
FC00 B6            OR   (HL)
             ;
             ;
FC01 2F      T1    CPL
FC02 32F75B        LD   (MSTR+7),A
FC05 FEFF          CP   255
FC07 2815          JR   Z,RESET
FC09 0604          LD   B,4
FC0B 21F05B        LD   HL,MSTR
FC0E 7E      LOOPP LD   A,(HL)
FC0F D3DD          OUT  (221),A
FC11 23            INC  HL
FC12 7E            LD   A,(HL)
FC13 D3DF          OUT  (223),A
FC15 23            INC  HL
FC16 10F6          DJNZ LOOPP
FC18 C3C105        JP   END1
FC1B D7      ERRP  RST  10H
FC1C F924          DEFW 24F9H
             ;
             ;
FC1E AF      RESET XOR  A
FC1F 32F05B        LD   (MSTR),A
FC22 32F15B        LD   (MSTR+1),A
FC25 060E          LD   B,14
FC27 21F05B  AGAIN LD   HL,MSTR
FC2A 7E            LD   A,(HL)
FC2B D3DD          OUT  (221),A
FC2D 34            INC  (HL)
FC2E 23            INC  HL
FC2F 7E            LD   A,(HL)
FC30 D3DF          OUT  (223),A
FC32 10F3          DJNZ AGAIN
FC34 3E07          LD   A,7
FC36 D3DD          OUT  (221),A
FC38 3EFF          LD   A,255
FC3A D3DF          OUT  (223),A
FC3C C3C105        JP   END1
             END   END

END    FC3F   AGAIN  FC27   RESET  FC1E
ERRP   FC1B   LOOPP  FC0E   T1     FC01
TONE   FBF6   NOISE  FBE1   ENV1   FBD9
ENV    FBCD   PERPL  FB89   P1     FB92
PLAY   FB83   LOOPS  FB76   C2     FB71
C1     FB59   ERRS   FB56   CHANS  FB34
PER    FB29   VS1    FB26   VOLS   FB1A
S1     FAFE   SOUND  FAEF   ERRM   FAEC
LOOP   FADF   OUTPT  FADC   CHAN   FABF
COUNT  FAB6   HALF   FAAF   OCT    FA9D
NOTE   FA81   VM1    FA7E   VM     FA72
M1     FA4E   MUSIC  FA3F   TABLE  FA27
START  FA08   INIT   FA00   MSTR   5BF0
STAKA  2D28   FINT2  1E99   FINT1  1E94
EXPT1  1C82   END1   05C1   STEND  05A7
ERR6   01F0   CHADD  0074   NXTCH  0020
GETCH  0018   *      5E18
```

Tortoise Wise

Lines from a parent who gets left behind. By David Stewart.

Oh Boy, is Winter really upon us? Is it nearly Christmas already? Will our telly screens soon be tempting us to part with our hard earned cash in order to fly off to sun drenched beaches and warm seas? Most of our friends are still haunted by the nightmare of the last Great British Summer. We'd been tempted this time last year and this summer we took the two boys to where the skies were blue, the sun shone every day, TV programmes were hardly ever in English and they couldn't get their busy little fingers on a keyboard or a joystick.

There were or course the usual painful withdrawal symptoms. Pale faces, vacant stares, twitching trigger fingers, waking us up in the middle of the night and pleading with us on bended knees to take them in search of the next day of a dark cool corner with a Spectrum and a program or two. But we remained adamant.

'Enjoy the scenery,' I said. 'Look, see that? It's the sun. It gives us daylight. Remember Daylight?' And they tried. They tried hard. With the most extraordinary results. . .

We had taken a book for each of them to ease the pain in those micro-less weeks. You may judge for yourself how desperate they had become by the third or fourth day away from their computer, when I tell you that they began to READ their BOOKS. That's right, a good old cover to cover, bleepless book. Real old fashioned PAPER and INK.

And, when they finished the first, they asked to be allowed to buy another, and then another, and another and we couldn't stop them. They took to books like people take to aspirin for hangovers. It was costing us a fortune in foreign currency and hours every few days to find a bookshop.

'Look,' I said in desperation once more. 'You will have to buy your own books with your holiday money from now on.' And they did, though not without protest. Not to me, but to each other.

'This is ridiculous,' said Number One Hare to his brother. 'We'll never be able to afford to read when we get back home.'

'Excuse me,' I interrupted. 'You could go along to the Public Library.'

'LIBRARY' they mouthed slowly as if the word were part *of a foreign language. I could see* they were searching their memories for data entered a long time ago on that part of Western Childhood.

'Oh yes' they sighed. 'You can borrow books there, can't you?'

I wonder, Tortoisewise, how many other words, concepts and expressions in this great language of ours are buried that deep in the vocabularies and experiences of today's young micro addicts? It's a thought for Tortoises everywhere. And that reminds me, thanks to all of you who have written and told me stories of parents who get left behind when it comes to their offspring and computers. Keep them coming I'll be featuring some in this column next time. ■

Harwood's Hypothesis

This month we have only four books appearing on this page. This could be due to the fact that publishers aren't letting us reviewers get our hands on the books and then beating them into pulp, or more likely it is because the computer book market is starting to falter. Will it eventually collapse?

The problem lies at the hands of two parties. The publishing industry and the computer owner. The former because they knew too early what a good thing was and rushed out and produced books to do this and that on all types of computer, completely flooding the market, which then slackens as everyone who wants one, has got one.

Another reason for the fall in the market is that all readers who wanted a book with lists of arcade games have by now got them. Then to produce a book on the Spectrum with lists of games after games is just pointless. Who would buy a new book when there have been thousands on the market for ages already?

That is why generally the books this month are starting to move away from the idea that people just want to program their machine to play space invaders.

Spectrum Wargaming Collins £9.95

The first book of the four is called *Spectrum Wargaming*, by Owen and Audrey Bishop and is published by Collins.

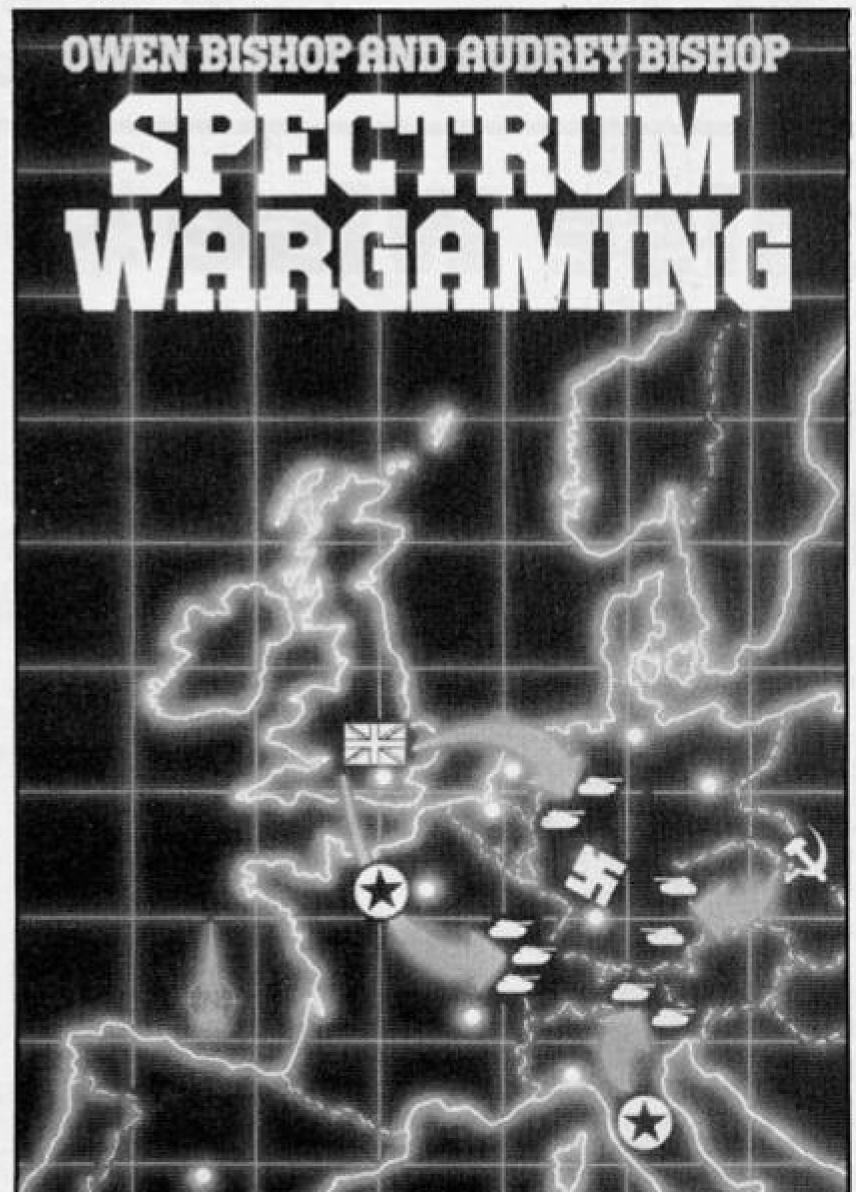
Wargames concentrate on tactics and the disposal and control of forces. Wargaming can be played either on a board, as in the usual board games, or more often a table, with models representing houses, soldiers, guns etc. You can achieve a high level of realism and accuracy in setting up the wargaming table, and this I am told, is half the fun of wargaming to its dedicated followers.

With a computer, it is possible, if you are not interested that much in converting the dining room into a battle field, to play the complete war game on the TV screen, although this method does lack the realism that most wargamers enjoy.

The book explains the principles of wargaming, from the setting of the scene, to the complex rules system. The book initially introduces wargaming, and then delves into the 'computer wargame', with a number of wargames listings.

It is possible, using the book to get the wargames up and running without any prior knowledge of computing. Four wargames are included in the book, so without any delay, apart from the laborious keying in, you can turn your Spectrum immediately into a war machine.

If you want a little bit more than just to play a war game, then this book is definitely for you. The authors have written routines that allow wargames to be tailor made, depending on the data that is entered for the game. Basically, the wargame program is very short, and is used as the controller of all the computer data. This data is the bulk of the game.



The wargame is moulded together by using various utilities. ie TABLEMAKER, ARMYMAKER and MAPMAKER. These utilities set up all the necessary wargaming data. The TABLEMAKER utility is used to set up the CRT's (not Cathode Ray Tubes but Combat Resolution tables), which are basically the standard rules of the game. eg number of turns per side, weapon usage, etc. ARMYMAKER is used to detail information about each side and MAPMAKER to describe the terrain in all areas of the battlefield. All these details amount to a vast amount of data, and because of this, the information is directly compressed into the bytes of the RAM instead of using BASIC variables. Variables take up on average 6 bytes, where we need use only one or less, by directly accessing the RAM. DATASTORER is used to compress all the data entered from the various utilities into the RAM, so it is easily accessed by a wargaming controller module.

The remainder of the book consists of the wargames. The first game is used only for example and does not make use of the ARMYMAKER utility, but only the TABLEMAKER. This is to give the reader a feel for wargaming, before jumping into it straight away. After doing this, the reader may enter the fully computerised version of the same battle, including maps of the terrain and German and French armies fighting against each other.

The other games included are a Napoleonic Campaign, a medieval battle and a conflict in outer space.

The authors have taken great lengths so that this book is adaptable to all aspects of wargaming, as well as using the full potential of the computer. Wargaming can be played between two Spectrums on an Interface connection, and details on how to achieve this are included.

The book is an excellent opener for anyone who wants to get into wargaming or even for the experienced 'manual' wargamer. The book is well written and takes the reader carefully through each section, so that by the end of the book, he or she will be writing their own war games, or at least adapting the ones in the book.

A very intelligent book, well thought out and enough to keep anyone occupied over the

winter months. Any wargame has the capability to be programmed, and this book provides you with the information you need. At £9.95 you may feel it is a bit expensive, but for the information inside, I would rather have this book than two games tapes any day.

Political and Military Simulation Games Interface £7.95

Remaining with the non-arcade idea, the next book is *Creating Political and Military Simulation Games on Your Micro*. Bit of a mouthful, eh? The book is written by Mike Rose and published by Interface.

The book is split into two sections. The first explains how to turn the initial idea into a computer program, and the second contains ten programs, illustrating the first section. Although the title only mentions political and military games, the book covers military, political and economic games. Perhaps there wasn't enough room on the front page to fit the whole title in! Military games come in two types. True military, which is the 'wargaming' as above, and then the strategic military. This is the type used in the book.

The reader is taken through step-by-step, starting with the planning of the game, and progressing till a complete model of the game is present. All aspects of the game are mentioned in the book and examples for each different group are given at each stage. This simplifies the idea, as it is much easier to learn something with three examples instead of one. By the end of the first section one should have a complete flow chart showing the structure of the game.

My first niggle comes at this point. In the introduction, we are told that the book 'requires only a relatively elementary knowledge of computing — only enough to put logical flowcharts into BASIC'. Why then is it necessary to flowchart the structure of a YES/NO question? This should appear pretty obvious to anyone with the slightest of programming knowledge.

Another chapter which is also not very useful is the debugging and trouble-shooter. The only thing this would be helpful for is NOT committing suicide. It says NOT to jump out of the



window if an error occurs!

The remainder of the book consists of ten programs, from a naval battle in Elizabethan Times to being a Prime Minister at 10 Downing Street (much the same I suppose?). These programs are well written and quite interesting to play, but nothing spectacular. I did like Nuclear Crisis, where the aim is to persuade the other countries to give you land.

In general the book is ideal for anyone with a knowledge of BASIC, and who wants more out of their computer than space-invaders. I did feel that at times the book expects you to know a lot more than at other times. But still, an interesting book, though perhaps slightly overpriced at £7.95.

How To . . . Babani Books

This book is the one that is contrary to my point in the first paragraphs, claiming that the reader is not interested in the usual games programs any more.

How To Write ZX Spectrum and Spectrum + Games Programs book is a step-by-step guide to writing games programs, and does at times assume the reader to be ignorant of computing, and at other times to know it all. On the first page, the invaluable REM statement is again explained, but at this point the reader is expected to know all about FOR . . . NEXT loops and the PRINT AT statement. The reader is also told how to get an asterisk on the screen by pressing Symbol Shift and B, but not how to get any of the other statements. Double Dutch if you ask me!

The step-by-step guide could be useful to the novice programmer, but perhaps the author thought that just a games book wouldn't sell too well, so the idea of learning how to program games seemed a better prospect. The programs are in separate chapters, and by the end of the first chapter the reader should know how to use the UDG's and the PLOT and DRAW statements. The games are separated into individual routines, and the author does his best to explain each routine, demonstrating how the game is moulded together. The games are at most for six players, but never included is the capability of playing with the computer. There is a chess game, but only the bare structure, as the CHESS game in the book is only for two players, and does not have the facility to compete with the computer. I would rather get the good old fashioned chess board out. This way is much more real.

The programs also did not appear to be all that well written. I could easily re-write a few sections if I was going to use the book to any great length. The listings also, due to the width of the book, were hard to read and appeared somewhat compact, but the programs could be adapted to improve the games, which in my mind only half succeeded.

At £2.50 the book is reasonably priced, and it may appeal to those who have some ideas but have difficulty in transferring them onto a computer, but otherwise I cannot really see the book as being extremely beneficial to anyone. Why would anyone want to draw a River Nile Scene in a game? Just another way of filling out a chapter.

Data Management at Work Pan Books £2.95

The final book is not intended for home computer users, but business users. The book does not relate to the ZX range of computers, but is ideal for anyone thinking of computerising their business, either at home or in the office.

The book is called *Data Management At Work*, written by Kathy Lang, and published by Pan. A the title suggests, the book introduces the reader to some of the capabilities of a business computer and the use of a Data Management System or Data Base System. A data base system is briefly a way of dealing with structured information (the simplest example is the card index system).

The author starts by presenting a problem in a social club, and the ideas behind turning the manual process into a fully automated computer system. The case study is at first *presented, and the situation* evaluated, so the reader can see examples of typical uses of a data base or management computer system. Before using a data base system and writing an application program, the manual method needs to be analysed, so that it is possible to see what the computer can and cannot do.

The book starts off by analysing the social club example, and turns this into a real-life situation explaining how the manual method can be computerised. The reader is taken through the various stages of the data base system, from the format of the records to the report structures that will be needed.

The author has gone into much needed detail, with the explanation of printers and the problems of setting up a computer system eg., How to train new users? Who should use the computer? etc. It is a well written manual to data base systems, and should be used as a forerunner to all businesses when the question is asked, 'Should we computerise?'. The book does not go into any technical depth, and is totally understandable to the non-computer mind, without being patronising.

Each chapter has an introduction, so it is clear what is going to be discussed in that chapter. There is also a short *precis* at the end of each chapter. Although the book is not designed to glorify any data base packages, it does suggest which ones may be most useful for your application. The correct system for the social club computerisation is then evaluated in detail, so it is easy to see how the method is applied when choosing the correct package.

This book is a must for anyone thinking about computing. If you have already followed the millions by buying a business computer, this book should also be read with interest to illustrate the capabilities of such a machine. This book is also one of the few that I can say is totally underpriced. At £2.95, everybody should have one! ■

ZX81 Chatterbox

Julian Chappel and his ZX81 ride again!

Greetings to all ZX81 owners once again!

I am very pleased to be able to report that, following the first appearance of ZX CHATTERBOX in the last issue of ZXC your letters have been flooding in. Both of them! At this rate we shall be completely bogged down with the sheer weight of correspondence by the year two thousand one hundred and ten.

Er. . . I have a very embarrassing admission to make, so I am writing this extremely quietly in the hope that this column has put in an appearance on these hallowed pages. I already have to apologise for a glaring 'boo-boo' in the first article.

I'm very, very sorry.

Right, having got the apologies out of the way we can get on with all the good bits. Don't listen to the owners of Spectrums and other lesser machines when they say that all the good points about the ZX81 can be written on the back of a postage stamp in large capital letters. They knoweth not what they are talkething about. After all, we knoweth different, don't we? Even with the advent of machines capable of printing 256 colours in an area the size of a flea's brain and sound good enough to put the London Philharmonic Orchestra to shame, the ZX81 still represents an excellent introduction to computing. It is no secret that the use of home computers is no longer the domain of the whizzkid. Many a whizzdad and whizzmum take an interest these days, but tend to be a little more cautious when entering the world of BASIC, binary and bugs than their offspring. Probably because it is their own money they are spending! It stands to reason therefore, that a ZX81 will have 'em programming like a good'un in less time that it takes to paint the Severn Bridge with a toothbrush and give a good indication whether or not further delving into the realms of computers seems attractive, all without parting with hundreds of pounds of hard-to-come-by cash. This brings us (rather neatly I think) to the first letter this comes from a newcomer to the ZX81 who was presented with it as a 70th birthday present! Beats woolly socks doesn't it?

(There will be a short delay before this letter due to a second apology. I am very sorry that I didn't tell you what the first apology was all about.)

Now back to the letter. . .

'Dear ZX Chatterbox
As per the Oct/November issue of ZX Computing magazine, I would like details of the S.F.S.C. having been a recipient of a ZX81 on my 70th birthday! Not used as much as expected as yet but winter is coming.

I have trouble (not unusual I understand?) with the LOADing program of various lengths both from commercial tapes and my own programs. What occurs is that apparently correct loading takes place as seen by the power meter, running to end but with no Report Code; as far as I know I have tried all ways of correction but to no avail.
Yours
W J Huggins, Slough'

At a time when the ZX81 is regarded as something akin to the Dodo it may seem strange that there are still a substantial number of people acquiring new ones. (Don't be silly I mean new ZX81s not new Dodos!). This means that there are many ZX81 users who missed the press coverage this little black wedge received during its heyday, and as a result are blissfully unaware of certain facts that us old hands simply take for granted.

Loading

One such fact is that the ZX81 is highly temperamental when it comes to SAVEing and LOADing programs. Another is that swearing at a ZX81 in a particularly obstinate mood has no effect whatsoever. (Except may be making it worse!) Hitting it with a hammer gives the advantage of relieving the user's pent up frustration with the offending machine, but also has the side effect of smashing it into little pieces. Not to be recommended. Here are some less drastic hints and tips towards reliable LOADing:

Setting up the machine

1. The ZX81 was designed for use with an everyday portable, mono tape recorder. A dedicated data recorder will be little or no better unless the facility to adjust the azimuth setting is incorporated. A stereo recorder (even a portable one) should be avoided, these may work but reliability will seriously suffer.

2. Clean the heads, rubber pinch wheels and drive spindle of your tape recorder thoroughly and regularly with cleaning fluid. Thoroughly dry and polish all areas with which the fluid comes into contact before re-inserting the tape.

3. Some recorders produce a feedback loop around the ear/mic sockets. Always remove the 'mic' lead when LOADing or the 'ear' lead when SAVEing. Make sure that the 'ear' and 'mic' leads do not cross (or are crossed by) either the mains cable to the tape recorder or power supply transformer, or the lead from the power supply. Separate all leads as much as possible.

4. Set the TREBLE control on your recorder to maximum and BASS to minimum (or TONE to high).

5. For reasons best known to Uncle Clive, the ZX81 is less temperamental when the program name is used. Avoid using LOAD" " whenever possible.

The signal

When LOADing, the ZX81 repeats all data read into memory out to the screen, giving those characteristic horizontal lines. This LOADing pattern can therefore give a useful indication of how the LOAD is progressing (but no more than that. It is by no stretch of the imagination an accurate or infallible pointer).

The black horizontal lines should be bold and clearly defined. They will be about 1-2 inches wide (on a portable TV) and the gap between each should be roughly equal to the width of the lines themselves. The width of the spacing will decrease with any increase of volume level. It is the definition of the lines themselves that gives an indication of uncertain LOADing. If they appear thin and straggly, or have a snowy effect (i.e. are flecked with a large number of little white dashes) and refuse to become clearly defined no matter what volume level, this is an indication of azimuth misalignment.

The 'business' part of the playback head is a microscopic slot, the angle of which in relation to the passing tape is critical. The ZX81 is probably the most sensitive to misalignment of any make of home computer. A good data recorder will have the facility to adjust this angle, usually in the form of a small hole positioned over the head into which an electrician's screwdriver can be inserted. Simply adjust the setting while a LOAD is taking place until the horizontal lines become bold and clearly defined.

Volume setting

Once again the ZX81 can be very sensitive to the volume level used when LOADing. Too low and the LOAD will fail. Too high and the computer can mistake the normal operating noise of the recorder as part of

the program, resulting in a 'bugged' copy ending up in memory. To find the ideal setting for any given program, try the following:

1. Start the recorder playing with the volume level at its LOWEST setting.

2. Type in LOAD''program name'' followed by N/L. Once you are sure the tape will be somewhere within the program itself, SLOWLY increase the volume until the characteristic lines appear on the screen. This is the lowest possible setting.

3. Rewind the tape and LOAD as normal at this volume level.

4. If the LOAD fails, increase the volume *very slightly*, rewind the tape and try again.

5. BE PATIENT. Don't try to hurry things up by increasing the volume in little jumps. By doing this the optimum setting may be missed.

6. Once a successful LOAD has been achieved, make a note of the volume setting for future reference.

7. If problems persist with any particular program it may help to listen to the beginning of the tape through the tape recorder's loudspeaker. Stop at a position just before the program starts and start LOADing from there. In this way any pops or clicks due to equipment being switched on or any other inadvertently recorded noise will be avoided.

There are two main requirments for regular successful LOADing with a ZX81. One is a portable tape recorder that incorporates an azimuth adjustment facility. With other computers you can easily get away without one, but it is virtually essential for the ZX81, especially when using tapes not recorded on your own machine. They are very easy to get hold of these days and are littel or no more expensive than an ordinary recorder. The second is a little harder to come by. I personally cannot find one single shop in the entire length and breadth of England that sells any, but it is probably more important than any other piece of equipment. It is patience. The patience to experiment carefully and methodically without resorting to swearing or hitting your beloved machine with a hammer. Neither of them work.

(There now follows a break for a third apology. I'm very sorry that I *still* haven't told you what the first apology was all about).

Beyond the pale

Before going on to the next letter it is important in the interests of safety to make sure that you are all in a sitting position. I would not like to be held responsible for any damage caused should any ZX81 owner faint out of sheer admiration for the bravery and courage of an intrepid member of the S.F.S.C. Against all the odds, and everybody's better judgement, he went out completely alone onto the streets of London, dispensing with the back up of an artillery battery and a couple of infantry divisions. Without regard for his personal safety he actually entered various computer retailers, walked up to the counter and asked if they stocked any ZX81 titles. If this alone does not astound you. If you have fainted out of admiration for his courage above and beyond the call of duty, then this might just do the trick.

He found some.

I will let him tell you exactly where. I'm sure you don't want to listen to any more of my drivel and I'd hate to steal his moment of triumph, so I hand you over to Simon Godfrey. (A round of applause please!).

'Dear Galactic Poultry

I'd just like to tell you how great I think your Club is and I can't wait for the new games mentioned in — (expletive deleted) — are on the shelves.

When I moved to London I decided to go and see what '81 software I could pick up. But, to my amazement, I found a considerable amount available. In Ealing (Nr Ealing Broadway, to be exact) try 'Armchair Action', on Oxford Street try 'McDonald's Hi-fi', which can be found near the Virgin Megastore. 'Low-Price Records, Cassettes, Software' on Tottenham Court Road sells top-class games such as pixel's *Trader* for £1.99, Bug-Bytes *Mazogs* at £2.99 and PSS's *Maze Death Race* at £1.99. To reach this shop, go down Oxford Street and when you reach Tottenham Court Road go in the left-direction-facing and after a small ealk you will see the shop on the other side of the road. Aslo along Tottenham Court Road is 'Sonic Foto Centre', who sell software and GASP! ZX81's and RAM packs. (The low-price shop sells '81s and keyboards for £10!).

In the north of England try 'Bradleys Records' in Sheffield and a really good budget software stall in Chesterfield's Friday open-market.

I would be very grateful if you publish this letter, as it will just help support the best value computer in the world.

Yours faithfully,

Simon Godfrey, London'

So what are you waiting for? Get that hammer you were going to use on your poor unsuspecting ZX81 and attack the old piggy bank instead!

(There now follows a short break. I am sorry that there will be no fourth apology, due to a shortage of supply.)

If you don't live near London, or around Sheffield then you will have to resort to writing your own programs. Here are a few tips from Jon Rose to help you.

'Dear Mr Chappell

Something you may be interested in to pass on to other ZX81 users is a few programming tips that I found whilst writing a program called 'Cabbages and Ants'. They are all to do with BASIC programming but, with sufficient know-how can be applied to machine code programs.

1. NEVER, NEVER use **RND** if you wish to get a random number in ZX81 programs. It is not random enough. Instead use **PEEK 16434**. This is the systems variable 'SEED' and seems to be very random, giving 230 different answers when **POKE**d in succession, where **RND** could only give fifteen without repetition.

2. To save memory use **INT PI** for three, **NOT PI** for zero or also **PI-PI**. Use **CODE** wherever possible. Use **VAL** on numbers above 255.

3. Want two bytes for some data? The addresses 16417 and 16507 aren't used.

4. Want to protect your REM lines?

(i) **POKE** 16514,118: **POKE** 16515,118

(ii) You might also like to try the code for 'cursor left' (114). This makes the REM line difficult to edit and annoys anyone wanting to cheat at your games.

(iii) To cancel the above out, poke with 0.

Jone Rose, Bognor Regis'

Well I think I've apologised enough. One can only eat so much humble pie before obesity sets in, after all. So now the mystery of the first apology will be revealed. Do you remember the 'jolly good program' in the last issue? You know, 'ZXYLOPHONE' by Paddy Moindrot? Did you try it? Did it work? No.

Ah. . . well, there is a very good reason for that. I forgot to mention that before RUNning this program you have to put your ZX81 into FAST mode. It won't work in SLOW.

Having already apologised several times you won't get any more sympathy from me, even if you have tried for two months to get it to work! Ha! Ha!

On second thoughts, before I leave you I just have to mention that the Software Farm is releasing two more fantastic programs for the ZX81. Just the thing for Christmas. As you have to be persuaded to part with your lovely loot perhaps I can manage just one more slice of humble pie. . .

If you would like a catalogue of ZX81 software, have any comment or query concerning the ZX81 however large, small, silly or serious then drop us a line at —

ZX CHATTERBOX
Software Farm
16 Charlotte Street
Bristol
BS1 5PP

Into The Archive

The QL's flexible database: by David Nowotnik

Of the four software packages to be supplied free with the QL, the database program, 'Archive' could well be the one most widely used. For while the QL rests in the precarious no-man's-land at the top end of the home computer market, and bottom of the cut throat business arena, information storage and retrieval will be the major link between the wide ranging band of QL users.

Whether one examines the original versions of 'Archive', supplied to all buyers of the QL prior to February '85, or the upgraded and improved 'V2.0', the overall conclusion is the same. 'Archive' can provide a superb database system. Its great strength is that it can be tailored exactly to meet the user's requirements; none of the compromises that accompany those ill-fitting, off-the-peg packages are necessary.

Archive's strength, however, is also its weakness. When loading Archive into your QL, you haven't got a working database, though you do have the means to create the database. Archive is a high level programming language, not too dissimilar from SuperBasic, which allows a database suite to be created. High level commands, such as 'Create', 'Insert', and 'Find' perform the majority of conventional database operations, but it is still left to the user to put together — to program — his/her lists of commands to create the required application for Archive.

Particularly in the business area, users might well be unable, or unwilling (for reasons of time) to develop their own suite of procedures which provide the basis of any database application program. Fortunately, there is no copyright restriction on individuals or software houses creating for sale database application software written in Archive's programming language, for use with Archive. While neither Sinclair Research nor Psion have any immediate plans to market Archive application programs, freedom from copyright limitations has allowed other software houses to produce packages (for example, Eidersoft sell a suite of 5 packages for £14.95). In fact, Psion have gone a step further to encourage application software. They have available for sale to individuals or companies a 'run-time' version of Archive. This program allows copy protection of Archive application software, making Archive a very attractive basis for companies wishing to develop and sell database systems for the QL. In the run-time version, the application loads immediately (no need to load Archive, then load the application). There is more RAM available for files, and there should be extra features not available to the standard version of Archive.

By buying one of these application packages, the user is, of course, restricting himself to an 'off-the-shelf' system, and so cannot fully benefit from the flexibility which is Archive's main strength. There is the option, for those unwilling to program Archive for themselves to pay a programmer to write software for them. But there is much to be said for trying to write your own software. The Archive programming language is much easier to use than BASIC, and the concepts of the language are adequately covered in the QL's User Guide, as well as a number of other books. The Sunshine Publication by Alison McCallum-Varey called 'Quill, Easel, Archive and Abacus on the Sinclair QL' is one good example of a text book which covers Archive programming in an easy to follow manner.

Your own database

If you are still not convinced that you could write your own database program, then read on. Writing computer programs, even in Archive's simplified programming language, is a matter of applying logic and common sense to solve a problem. Reasonably, assuming that everyone is blessed with some abilities to think logically and use common sense, programming Archive should be within anyone's capabilities.

Setting up a database system should involve a preset sequence of tasks; defining exactly what is required, planning a logical sequence of operations to meet the objectives, then writing procedures which carry out these operations. The procedures are tested individually (to iron out bugs), then a linking procedure written to pull together the whole sequence.

To demonstrate Archive's simplicity, here is a simple database application, a name and address file. The list of procedures which make up this file is shown in fig.2. But what may be more relevant to the budding programmer are the processes which were involved in designing the system.

When analysing any potential computer application, the first question that must always be asked is 'do I really need a computer to do this?'. In the case of a name and address file, if all that is required is to look up an address or telephone number, using an address book will always be much simpler (unless the names exceed several thousand!). One justification for a computer file is the need to sort the records in more than one way. I find that I have a remarkable ability for forgetting birthdays, with the result that family and friends are often left disappointed. So this name and address file includes the ability to sort through the records, and identify those families in which a birthday is imminent. The records in the file are in no particular order (but this could be achieved with simple modification using Archive's

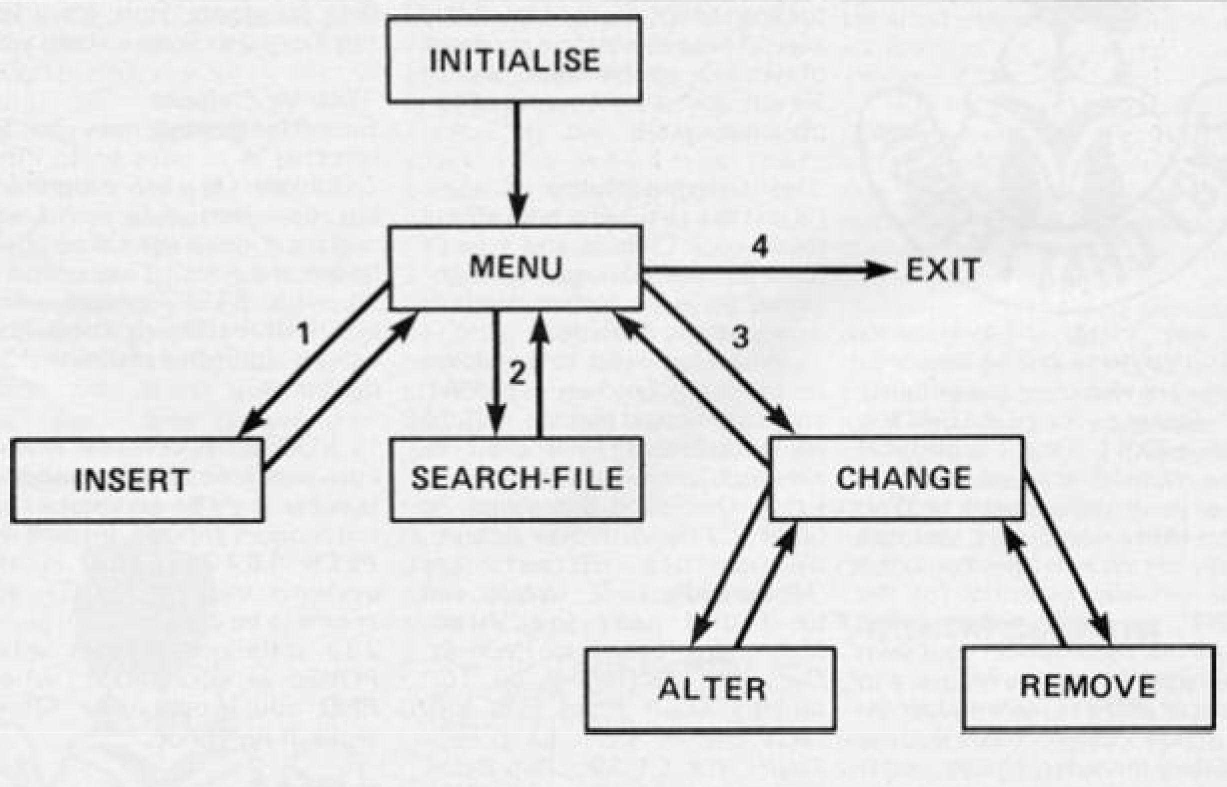


Figure 1: Schematic layout of the address program

powerful language); instead the flexible searching facilities of Archive are used to find relevant records.

With the twin needs of the filing system identified, the system can then start to be designed. Each record should then contain surname, christian name(s), the address, the telephone number, and space to list birthdays. This file allows up to three dates per family. File creation with Archive is a simple matter of typing in the series of direct instructions, starting with the command 'create' and ending with 'endcreate'. It is even simpler with the 'address' program. Instead of using a series of direct commands for file creation, the name and address program in Fig.2 has a special procedure called 'make_file'. All you have to do is, with the program in place, type the direct command 'make_file', and a new file will be created on microdrive 2.

A simple database program will require a minimum set of operations to be available to the user. These are:

1. Add another record
2. Search for one or more records
3. Modify or delete a record
4. Exit from the program

These options are available most conveniently through a menu. The options are usually numbered, and pressing the appropriate number key allows you to inform the computer of your selection.

This is how the name and address file works. Take a look at Fig.1 to see how it fits together. When the program starts, it should go into an initialisation procedure which sets up the program, ready for use. In this case, the procedure called 'initialise' sets a 64 character width mode and opens the address film on microdrive 2, ready for read/write operations. The program then moves into the menu which prints the options onto the screen. After this, the program waits for a key press, and responds only if keys '1', '2', '3' or '4' are pressed (the 'if' statements). With the appropriate key press (1 to 3), the required procedure is called. If '4' is pressed, the file is closed, and the program is terminated.

Options

Fig.1 demonstrates how, with options 1 to 3, the program

Figure 2. Address program listing

```
proc change
  seek
  print at 10,10 "Delete or modify? (d/m)"
  let t$="a"
  while not (t$="d" or t$="m")
    let t$=getkey()
    endwhile
  find a$
  while found()
    display
    sprint
    if t$="d":remove: endif
    if t$="m": alter : endif
    continue
    endwhile
  endproc
proc initialise
  mode 1,6
  open "address"
  endproc
proc make_file
  create "address"
    surname$
    firstname$
    street$
    town$
    county$
    telephone$
    name1$
    birthday1$
    name2$
    birthday2$
    name3$
    birthday3$
    endcreate
  endproc
proc menu
  let s=0
  while s=0
    title
    print at 5,5;"1. Enter new record"
    print at 7,5;"2. Search file"
    print at 9,5;"3. Delete/modify an entry"
    print at 11,5;"4. Exit file"
    print at 13,24;"SELECT 1,2,3, OR 4"
    let s=val(getkey())
    if s=1: insert : endif
    if s=2:search_file: endif
    if s=3:change: endif
    if s=4: close : endif
    if s<>4: let s=0: endif
    endwhile
  endproc
proc remove
  print at 14,5;"Do you want to delete this record? (y/n)"
  yes_or_no
  if p$="y"
    print at 14,5;"              Are you sure? (y/n)            "
    yes_or_no
    if p$="y": delete : cls : return : endif
    else
    print at 14,5;"                                             "
    endif
```

```
   endproc
proc search_file
   seek
   find a$
   while found()
      display
      sprint
      print at 14,10;"PRESS ANY KEY TO CONTINUE"
      let q$=getkey()
      print at 14,10;"                         "
      continue
      endwhile
   endproc
proc seek
   cls :title
   input at 8,10;"Enter a search word - ";a$
   endproc
proc start
   initialise
   menu
   endproc
proc title
   cls
   print at 1,22;"NAME AND ADDRESS FILE"
   endproc
proc yes_or_no
   let p$="a"
   while not (p$="y" or p$="n")
      let p$=getkey()
      endwhile
   endproc
```



cycles round back to the menu, ready for the user to make the next selection. For programming purposes, option 1 (enter a new record) is the simplest to incorporate. It uses the Archive command 'insert'. With this in-built producedure, one or more new records can be entered. Clear instructions at the top of the screen, generated by 'insert', makes the process of data entry very simple to do.

The second option in the menu, Search file', allows you to find any record which has a certain string of characters anywhere in that record. The procedure 'search_file' firstly calls another procedure 'seek' which asks you to enter the string of characters for search purposes. The Archive command 'find' is then called, which goes through the file until an exact match is found with the characters you entered. For search purposes, upper case letters are the same as lower case. Once a match is found, then the 'while. . . endwhile' loop is brought into play in 'search_file'. It displays the record on the screen. The program pauses, let q$ = getkey ( ) while you extract the infomation you want, then the search carries on (the Archive command 'continue') until more matches are found, or the end of the file is reached. At this point (and if no match is found), the procedure ends, and the program returns to the menu.

You can search for whatever information you want with the flexible 'find' command. For example, if you want a family record displayed enter the surname. If you want all the birthdays in September, enter 'Sep', or whatever abbreviation you use for September when inserting records.

The third menu option allows you to alter a record, or delete it. Within the 'change' procedure, the 'seek' procedure is again called. After all, you first have to find the record you want to change. Then you are asked if you want to delete or modify a record, to which you should respond 'd' or 'm'. Having made these selections, Archive's 'find' command is again used to search through file to find the appropriate record(s).

Once found, the record is displayed; if you asked to modify records(s) then the convenient Archive command 'alter' is used to allow you to make changes. 'Alter' works in much the same way as 'insert', and all the instructions needed to make changes are on the screen. If you don't want to make changes, simply press 'F5'.

If you requested 'delete', then the 'remove' procedure is called. This asks you twice (to make sure) if the displayed record is the one you wish to remove. If you press 'y' twice, then the Archive command 'delete' is called to remove that record from the file.

Going loopy

If you have followed the program description while reading through the program listing, how the program works should be reasonably clear, even to the non-programmer. Probably the most dificult concept for the novice is the loop — 'while. . . endwhile'. This may need a little practise, but the QL User Guide covers this rather well, so it should cause little problem after some practise.

To enter the listing into Archive, type in the direct command 'edit' when the " appears at the base of the screen after booting (loading) Archive. This gives you the word 'proc'. Type in 'change', press ENTER, then enter each program line in sequence, until you come to 'endproc' in the listing. You don't have to type in 'endproc', as this is provided automatically. Nor do you have to worry about the indentations in program lines, as this is also done automatically.

To enter the next procedure, press 'ESC', then 'F3', then 'n'. This allows you to start another procedure. And so, carry on in this way until all the listing is in place. Then save it on microdrive 2. Escape from the edit routine by pressing 'ESC' twice. Then type in save, and press 'ENTER'. This gives you some quotation marks, so place your program name (eg 'address') in the quotes, and press ENTER. The program will be saved on cartridge.

To load the program (the next time you want to use it), type run "address" after booting Archive. The program will be loaded and run automatically. This autorun facility will search for a procedure called 'start', and one is provided, which combines the initialise and menu procedures. Hopefully, this short Archive program listing will demonstrate how simple it is to write your own Archive application programs. Archive has the flexibility to be extended from such a simple one file application, to quite sophisticated multi-file systems. ■

QL Software

Damien Clay takes a look at some new games for the QL.



QL Hyperdrive
English Software
£14.95

This is the sort of game where you have to imagine you are a Formula 1 racing car driver and you have to dodge past each car to get into a suitable position (ie 1-5) to get to the next stage. There are five stages and each one is more difficult than the last.

The hi-res 3D graphics are very good, flicker free and look very realistic with good use of colour. The sound is also very realistic, and gives you the feeling that you are actually driving.

It is fairly easy to backup onto disk with the Sinclair utility disk supplied with their drive and, if you know a little about programming, to another microdrive. There is no backup program on the cartridge itself though.

Playing the game is fairly straightforward, all you have are four controls: left, right, accelerate and decelerate, using the cursor keys. It is up to you what speed you go, but the slower you go the less likely you are to get onto the next stage and the faster you go the harder it is not to crash.

The game is presented very well and displays the instructions on which keys to use, it also demonstrates parts of each stage. It comes in a wallet type case with space for another microdrive cartridge. There is a high score routine which displays the highest score but there is no hall of fame.

Overall this is a very good game and is very good value for £14.95 considering that English Software provide a lifetime guarantee on all their games.

Graphics	★★★★★
Addictiveness	★★★
Overall	★★★★

Cuthbert In Space
Microdeal
£14.95

This is a typical arcade game where you, as Cuthbert, have been sent on a looting mission in outer space but before leaving each planet you visit, you must use your remote shuttle to collect 15 fuel pods. The colour and the graphics are very good and hardly flicker but there is a lack of sound, all you get is a few bleeps, which does not make it very realistic.

It is very easy to make a backup copy as there is a backup program on the microdrive cartridge.

Playing the game is quite simple though the actual movements of the character are quite fast. It is fairly enjoyable and also reasonably addictive. It is presented very well in a converted cassette case which unfortunately does not hold a backup tape. The program does not give on-screen instructions but these are provided on the cassette inlay card.

There is a very simple high score routine but there is no hall of fame, which is a pity because you can't put your name up when you get a good score. Overall this game is very good but a bit expensive, however with the lack of QL software, if you have the money to buy it, it is quite enjoyable.

Graphics	★★★★
Addictiveness	★★★
Overall	★★★★

QL Cavern
JMF/Sinclair
£14.95

This is one of the best arcade games I have seen for the QL and is very well set out. Your objective is to travel the length and breadth of the cavern in order to collect all 395 diamonds.

Throughout your journey you will have to walk, crawl, swim, travel by boat and fly by jetpack. The colour and the graphics are very good and flicker free and the sound, though limited, is very good.

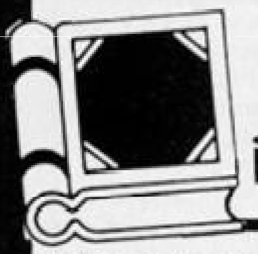
The game carries a backup program on the microdrive cartridge and instructions are in the little booklet with extra information given onscreen from the start menu.

Play is very enjoyable and addictive, some screens are difficult, some are complex and some are simple but they are all excellent and very exciting. It is presented in a wallet type case with space for a backup microdrive cartridge. The instructions are clear and easy to understand as well as being brief and well laid out.

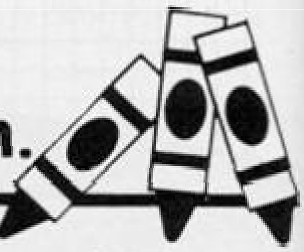
There is no high score routine which is a pity because it does tend to spoil the program a bit. Overall it is a very good program and has been well designed and programmed. I personally think it is worth every penny and is well worth adding to your collection.

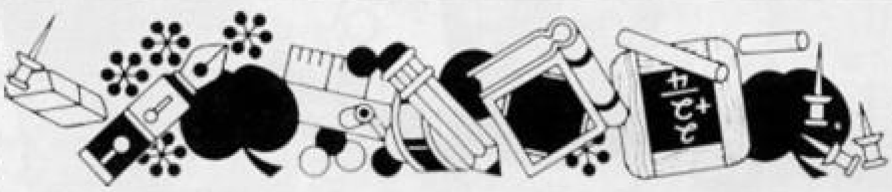
Graphics	★★★★★
Addictiveness	★★★★
Overall	★★★★★

Spectrum Education



Mike Edmunds grades a couple of packages for those interested or involved with computers and education.





Sitting in a tiny bedroom you look around and gaze with horror at the mess surrounding you. Pieces of paper with hastily written notes, formulae and jottings cover the carpet and overflow from the wastebasket! A cold sweat beads your brow as the enormity of the task facing you finally sinks in . . . . . !

No, it's not the setting for a new adventure, merely memories of schooldays with examinations looming! Did you, like many others, find the task of revision a daunting prospect? I certainly did and, on more than one occasion, remember rushing to the nearest branch of a certain high street bookseller to search the shelves for those little packets of revision cards!

I doubt that the task of cramming for exams has become any less onerous but at least, in this age of the computer, you can test yourself with a whole range of programs which, to a certain degree, allow more flexibility than a textbook with the added benefit of less pieces of paper!

Letts, the producers of those aforementioned little cards, have also branched out into the software scene and have now introduced a computerised version of their Keyfacts for students of O-Level, GCE, CSE and GCSE Chemistry. At £11.50 the suite of programs is not cheap, but you do get nine programs for your money. The programs have been designed to test your knowledge and to develop your deductive facility.

The suite covers the following areas; Atomic structure and bonding, Periodic tables, Formulae, Chemical deductions, Electrolysis, Apparatus, Acids bases and salts and Organic Chemistry. The programs come on two tapes in a substantial wallet together with a pamphlet outlining the programs and a suggested list of further study aids (including little packs of cards!)

The programs themselves are equally substantial in terms of what they actually contain, but my overall impression is that the package has a rather 'bitty' approach. This is not to denigrate the actual Chemistry content which seems to cover well the areas concerned, it is merely the way that the whole thing has been presented.

The presentation, I suppose, is not that vital when you are only concerned with the facts and exercises, but it seems rather unfortunate, for example, to go to the lengths of a specially designed title screen only to have it disappear after a few seconds while the rest of the program loads to an empty screen, loading time is long enough as it is!

Programs

However, on to the programs themselves. The first program, Atomic structure and bonding, is one of several that use a comprehension approach to learning. A passage is presented and the user then has to fill in the missing words, much in the style of the standard 'cloze' procedures. Now this is not necessarily a bad approach, but when half of the words have little relevence to the chemical aspect of the passage I can see no reason to adopt this approach. Words such as 'the', 'and' etc. certainly did not test my knowledge of Atomic bonding.

The passage, (there is only one) is subject to five levels of difficulty, the harder the level the more words missing. At the hardest level I can see some point, but at the easiest it tests very little except your eyesight (cyan text on white paper!)

The 'Periodic table' program tests your knowledge of the location in the periodic table of some of the more familiar elements. Alternatively, you can select options which require you to apply your understanding of the electronic configuration of atoms.

'Formulae and equations' requires you to determine reactants and products. You also need to complete equations or organize a selection of chemicals to produce a valid balanced equation. This program is awkward at first, as subscript numbers have been asigned to graphics keys, this is rather fiddly, as are some of the other inputs required, but soon grasped.

'Chemical deductions' tests your overall knowledge of Chemistry by supplying clues about the nature of a substance which has been randomly selected. In 'Electrolysis' a random combination of electrolytes is selected and you are required to identify the ions present, say which electrode they migrate to, identify the preferred reaction at each electrode and compile the ionic equation for each electrode reaction.

'Apparatus' puts you back into the lab selecting the correct apparatus for the preparation of thirteen gases. Your selections are examined and the correct apparatus is drawn. You are then required to provide a balanced equation.

In 'Acids, bases and salts' you have to complete descriptive passages and examine given reactions for validity. You can also see how pH levels change during acid-base titrations.

The 'Organic Chemistry' programs are in two parts, one being the missing word passage, at the end of which the content is represented diagramatically. The second requires you to identify isomers, homologues and compounds with similar reactivity. Chemistry being the complex animal that it is I make no apologies for the descriptions above, these being necessary to give a fair picture of the programs.

As a package this covers almost everything that the exam student is likely to require. Graphics, where applicable, are competent and the complete suite goes a long way to making the subject more approachable and enjoyable. Although competent I feel the overall presenta-

tion is lacking in some areas. There is also a bug which causes one program to crash and obviously needs sorting out!

This is not a cheap package, and there are several programs available at present from other companies which do the job equally well, but if you have a need for a revision package then this could go very well with those little packs of cards!

Software directory

Now to software of a different kind, that contained as entries in a Software Directory, from Macdonald and Evans Ltd. The directory is arranged according to subject areas and is designed to provide all types of educational establishment primary, secondary and beyond — with up-to-date and concise information on educational software currently available.

The entries appear alphabetically by software title, and provide a guide to the age group where possible. Each entry indicates the machine which the software runs on and the form it takes. Prices are included though purchasers are advised to check details with the suppliers prior to purchase. The whole range of curriculum subjects is covered and the two largest sections, Mathematics and English, are divided into broad age-groups. As well as subject software the directory also gives an extensive bibliography of educational computing references, together with lists of periodicals, computer organisations, computers for Education and a list of software suppliers whose software is included within the directory.

The Directory is certainly comprehensive but, as with similar books covering other areas, it suffers in that it seeks to embrace too large an area. The age range, beginners to secondary and beyond, is just too large to be of use to specific users, say in a Primary School, who would look to other sources of information. The major drawback with something of this nature is that, like micros themselves, as soon as it is released it is out of date. There are notable omissions in certain subject areas, software that has been widely reviewed elsewhere, but, then again if they don't send details . . . .

These omissions illustrate the speed with which such publications become dated. Many programs remain a mystery, there is little or nothing in the directory to indicate their educational nature. I know what Dart and Logo do but there are no details within the book for those who do not. Similarly, several of the organisations listed are not adequately described.

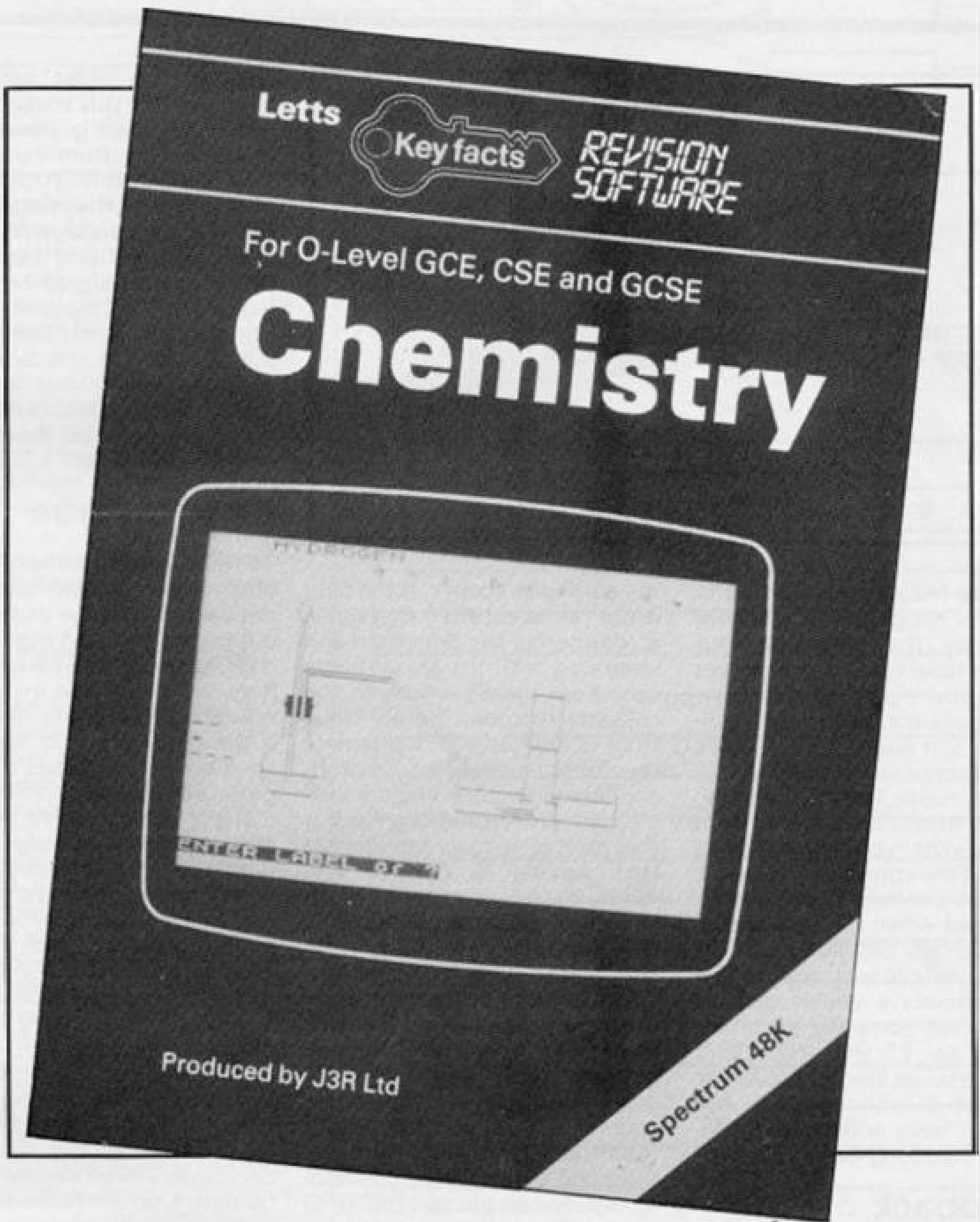
This can hardly be called a fair review of such an extensive book, that would require much more space than is available here. I would suggest that if you have a need for a software reference book then this is the best that I have currently seen, though I personally am currently using some excellent programs that are not even mentioned here.

The Directory is certainly very comprehensive and informative, (it is intended to update it at appropriate intervals) and overall this is a valiant effort to make the educational field aware of the range of software currently available, but for me it attempts too many areas for too many machines all at once!

That's all for now, next issue I'll be looking at some adventures especially for children.

Chemistry,
Letts Keyfacts Revision Software,
Spectrum 48K £11.50
Charles Letts & Co., Diary House, Borough Rd., London SE1 1DW.

The M & E Educational Software Directory
J. Arthur & T. Russell
Macdonald and Evans Ltd. Estover, Plymouth PL6 7PZ (covers most micros) £12.50.



Safe Save Booster

Anthony Marlow from Cheshire has sent in a useful little device for helping with SAVEing programs

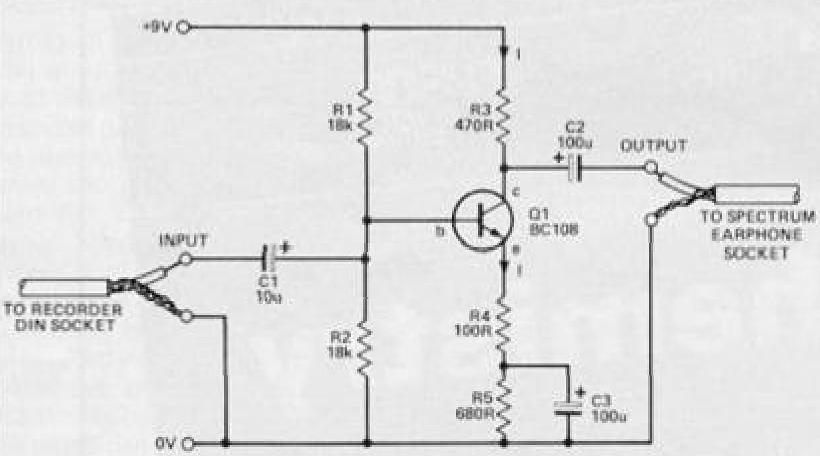


Fig. 1 Circuit diagram of the amplifier

If you're fed up with starting to SAVE a program on cassette and only then realising that you've forgotten to disconnect the earphone plug, or with exerting a considerable force in pulling out and pushing back in the power supply plug to provide a system reset, then this simple, inexpensive hardware project may be what you're looking for. None of the components used is expensive or hard to get hold of, and even when put in a plastic case will only cost about £3.50. This is much less than some commercially available designs which just comprise an on/off switch for £5.00. You don't have to be an electronics boffin to build it, although you will need to have access to a fine-tipped soldering iron.

Feedback

The need to remove the earphone plug while saving a program stems from the fact that most domestic portable cassette recorders feed the signal that is being recorded to the earphone socket. If the plug is not removed, then this signal is returned to the Spectrum and interferes with the signal being sent from the Spectrum to the cassette recorder. So a vicious circle is created which prevents the correct signal being recorded. Obviously, one could install a switch in the earphone line but the result is no more convenient than having to remove and reinsert the earphone plug. However, a number of cassette recorders also possess a 5 pin D.I.N. input/output socket, which is normally used for recording between two cassette recorders. A similar feedback problem could also exist between two cassette recorders and for this reason the output from the DIN socket is normally switched off inside the cassette recorder when put into RECORD mode. Thus, if use could be made of this output rather than the earphone socket, our vicious circle will be broken automatically without us having to unplug or switch anything, unlike many commercial devices. On most recorders, however, the DIN socket output is not sufficiently powerful to drive the Spectrum input when loading a program. To overcome this problem the simple transistor amplifier stage (Figure 1) was designed and built. This amplifies the signal by nearly five times, so that the output signal is capable of driving the input circuitry of the Spectrum. Note, however, that on some machines the DIN socket output is independent of the volume and tone controls.

How It Works

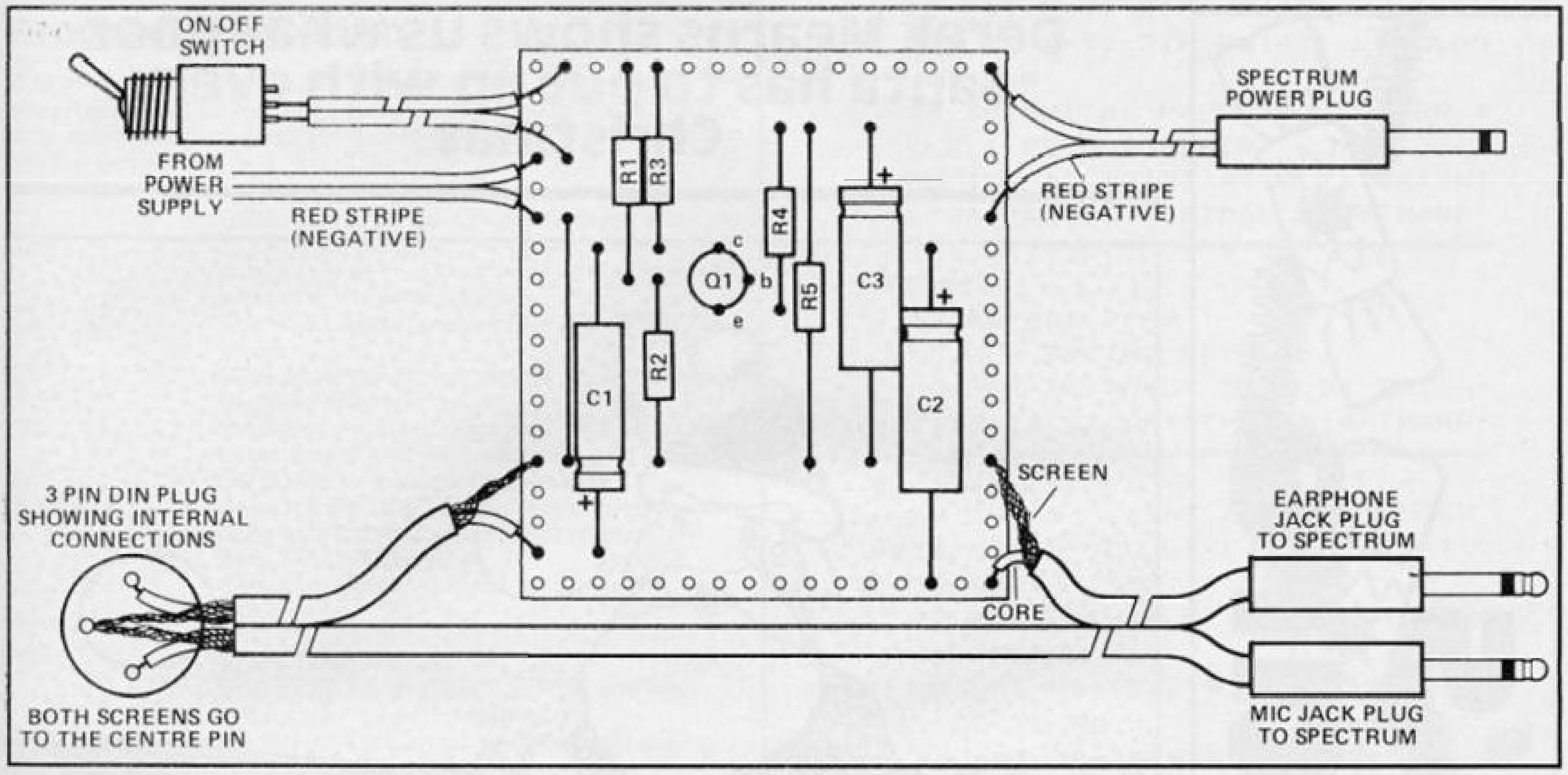
To fully understand how the circuit works would require a detailed knowledge of semiconductor theory. However, a brief description of what a transistor does will give some insight into how the circuit works. Basically, a transistor (TR1 in our case) consists of two pieces of an impure semiconductor material (usually silicon) with a thin slice of the same semiconductor (but with a different impurity) in between them. The two outer pieces are called the "emitter" and "collector" and the thin slice in the middle is called the "base". These are labelled as e, c and b respectively on the diagram (Figure 2a).

One of the properties of a transistor is that a small variation in voltage applied between the base and emitter (the signal) can cause a large variation in the current, I, which flows from the collector to the emitter. Thus, if a resistor is placed in the collector lead (R3) then by Ohm's Law (V = IR) the voltage at the collector will be proportional to this changing current, the precise value being determined by the value of R3. By choosing appropriate component values in the circuit, any desired amplification may be achieved.

Before the transistor can do anything, however, certain static d.c. conditions have to be set up. R1 and R2 form a potential divider which sets the static base voltage. Because the values of R1 and R2 are high, the best voltage can still be altered by a small amount when a signal is applied to it via capacitor C1. The static base voltage in turn sets the static emitter voltage. Given this, resistors R3 and R4 set the necessary standing current which flows between the emitter and 0V. The gain, or amplification, of the amplifier is set by the ratio R3/R4. Capacitors C1 and C2 allow the signals to pass through but prevent the standing voltages on the transistor getting through and upsetting the d.c. conditions in the recorder or Spectrum. The circuit takes its power from the Spectrum power supply. Since this involves cutting the wire from the power supply, a switch was added at the same time so that power could be switched off and on to effect a system reset.

Construction

The prototype was built on a small piece of 0.1" matrix Veroboard, available from most electronic component suppliers. The actual size required is 16 holes by 18 holes with the copper strips running along the shorter dimension. This can be cut from a larger piece. The component layout and connections are shown in (Figure 2a). Begin by fitting the resistors on the plain side of the board. Pass the leads through the appropriate holes and solder them to the copper strips on the other side. Use only fine multicored solder *and make sure the solder does not bridge the tracks*. Trim the leads close to the soldered joint. Next, fit the capacitor and transistor in the same way, taking great care that they are inserted the right way round (the capacitors will be marked as being positive or negative at one end and the tag on the transistor case is adjacent to the emitter). Now take a length of stereo screened cable (about half a metre) and carefully cut one wire about eight inches from one end, leaving the other wire intact. Strip the outer insulation off both cut ends and separate the screen wires from the central core, then remove about a



quarter of an inch of insulation from the core. Solder the two cut ends to the board as shown in (Figure 2a) with the shorter length to the right. Two 3.5mm jack plugs are now fitted to the shorter length of the stereo cable. Separate the cable by a few inches before fitting the plugs and remember to thread each plug cap on the respective half of the cable *before* soldering the wire to the plug!

In each plug the cable screen goes to the outer connection and the core to the central contact. Grey and black jack plugs can be obtained to match those on the original lead. The DIN plug is next to be wired up. Only a three pin plug is required even though the cassette recorder will probably have five or more pins. For details of the connections refer to (Figure 2a). Next, cut the power supply lead about five inches from the Spectrum plug. Strip the ends of the wires and connect them to the matrix board as shown in (Figure 2a). There's a red stripe down one of the two wires and, strangely, this was found to be the NEGATIVE side of the supply. Incidentally, it seems to be quite common for one of the wires to break inside the moulded-on plug due to flexing of the wire as the plug is pulled out and pushed in. Rather than sending the whole power supply unit off for a costly repair, the plug can be cut off and replaced with another. It's just a low voltage d.c. connector, widely available for just a few pence. The negative (red) wire connects to the inside of the plug and the black to the outer part.

The only remaining connection to make is that between the board and the on/off switch. A short length can be removed from the power supply lead when cut for the purpose. Any single pole single throw (S.P.S.T.) switch can be used. Just connect the wire as shown in (Figure 2b). The orientation of the wire is not important. Finally, the prototype was installed in a small plastic box for neatness.

The Circuit in Use

Provided the circuit has been built correctly, it should work first time. Another jack plug will be needed for the cassette earphone socket in order to switch off the loudspeaker when loading a program. Alternatively, if an earphone is plugged in, then it will be possible to hear a program being saved. If the circuit does not work, it is likely that a mistake has been made in the construction. Make sure there are no solder bridges across the tracks and that the capacitors and transistor are inserted correctly. The circuit has been found to slightly improve the ability to load some programs, but cannot improve what isn't there; most loading problems are due to substandard recordings and badly misaligned tape-heads. It will work with the new 'turbo-load' method now used by many companies.

List of Parts

2 × 18K ¼W Resistor (R1,R2).
1 × 470Ω ¼W Resistor (R3).
1 × 100Ω ¼W Resistor (R4).
1 × 680Ω ¼W Resistor (R5).
1 × 10μF 16V Electrolytic Capacitor (C1).
2 × 100μF 16V Electrolytic Capacitor (C2,C3).
1 × BC108 Transistor (TR1).
1 × S.P.S.T. Switch.
1 × 3 pin DIN plug.
3 × 3.5mm jack plugs.
1m Stereo Screened cable.
0.1" Veroboard as required (see article).
Box as required.

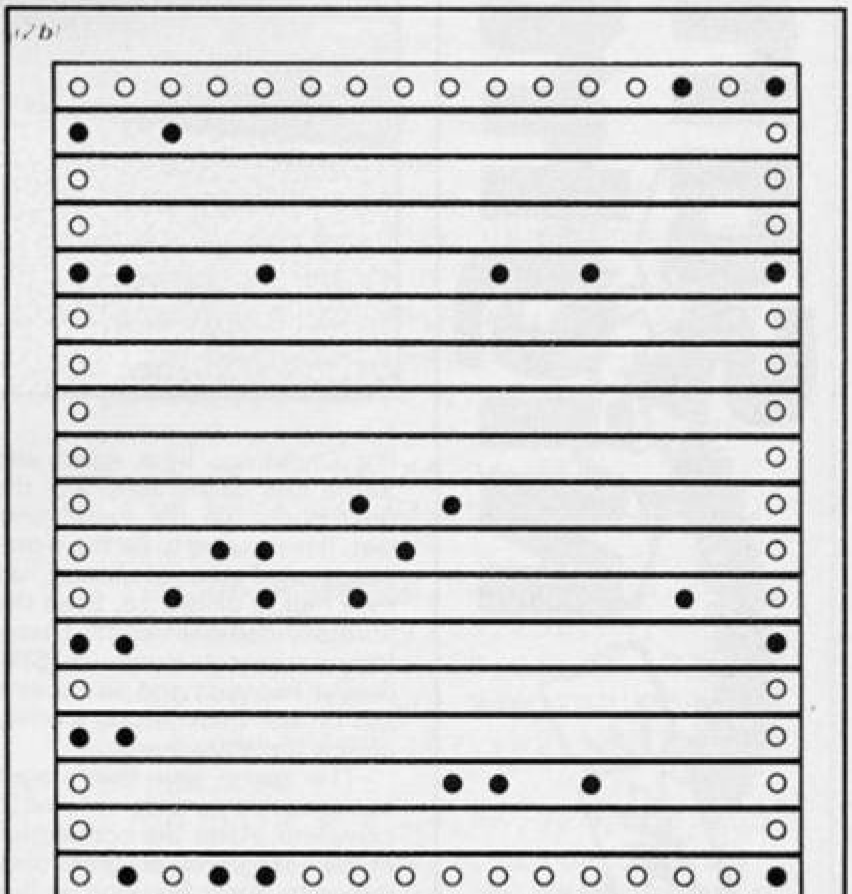
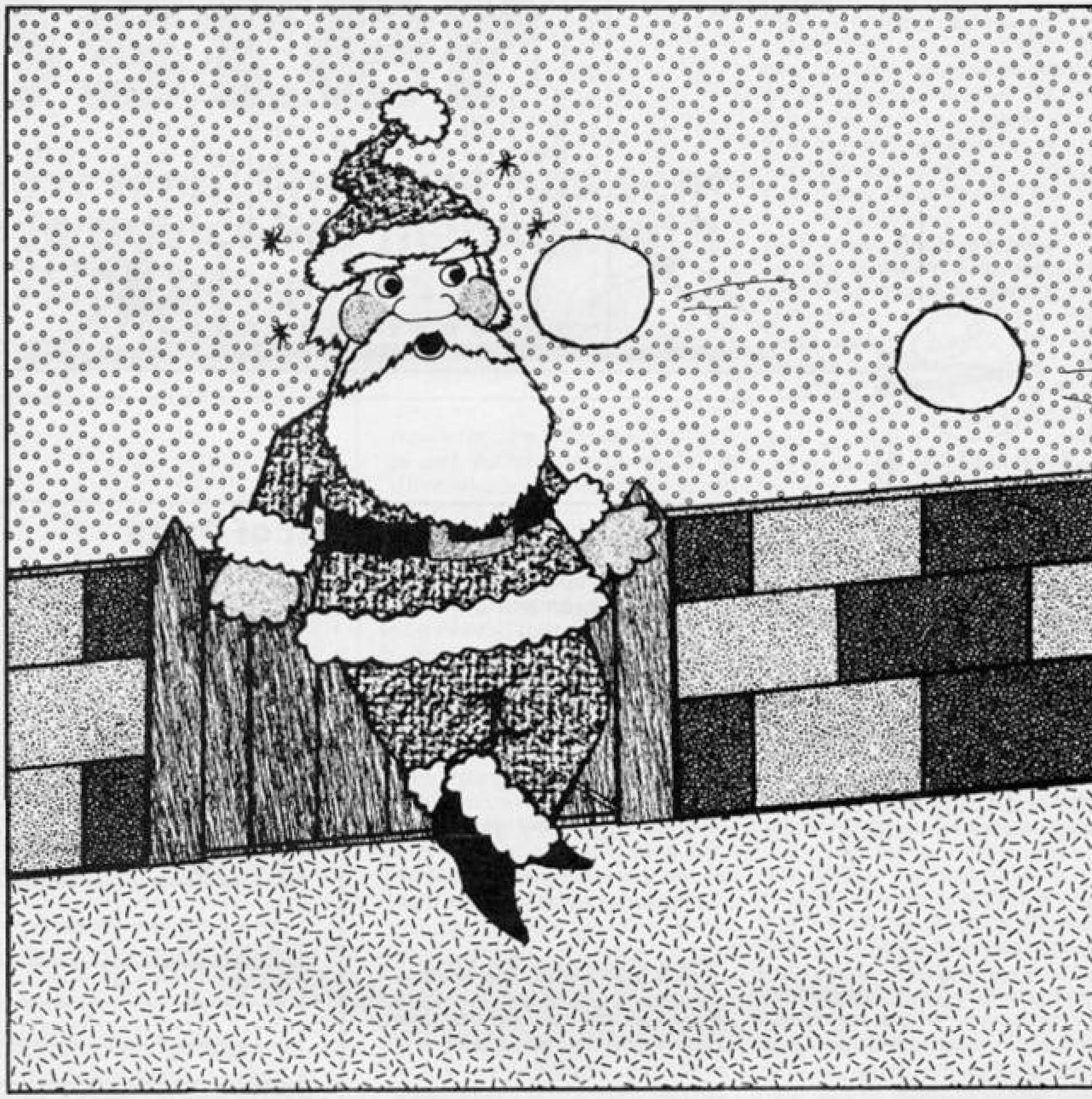


Fig. 2 Construction, showing layout of components and wiring (above) and solder connections on the underside of the board (below)



Derek Mearns shows us what poor Santa has to put up with every Christmas.



It's Christmas time again and Santa has deliberately left the next house on his round until last. It is reputed to be the worst house any Father Christmas has ever had to deliver to. Even the normally compassionate Easter Bunny curses its existence! Still, deliver he must and so he sets out on the treacherous journey across the snowy artic.

The game has five stages each requiring a different skill to complete. After the completion of each stage you progress onto the next. After finishing all five stages at December 1985 you progress to Christmas the next year and then to 1987, 1988 and so on with each year being more difficult than the previous one.

Variables

a,b	reserved for READing DATA.
e	stage.
f,g,j	reserved for FOR . . . NEXT loops.
h	high score.
l	number of lives remaining.
p	print positions for the obstacles on stage one.
r	reserved for RANDom numbers.
s	score.
t	various including time on stage 3 and total of key presses on stage 5.
u	Flag for whether or not Santa is Up in the air on stage 4.
x	X Co-ordinate of Santa.
y	year.
c$	width of roof and Chimney on stage 2.
h$	name of the high-scorer (maximum 10 letters).
k$	contains the user-defined keys.
r$	reserved for INKEY$;

Program Breakdown

A brief program is included in the form of REM statements within the program but for those of you who are interested what follows is a more detailed program including notes on what certain special commands (e.g. the POKEs) do.

15	GOSUBs the UDGs. Originally these were part of the main program but were changed to a subroutine so that they were set up to be printed in the instructions.
20	The POKE sets the computer in CAPS LOCK mode.
25	Sets up variables that should not be reset at the start of a new game.
30	Initialises variables that need defining at the beginning of each new game.
52	GOSUBs the main instructions.
60-99	Asks for the User Defined keys. Enter can be used as a key but will present problems in lines 9300-9550.
900-995	Sets up variables needed for stage one including: 910 P is set up to hold the print positions and types of obstacles. 950 GOSUBs the NEXT STAGE and brief instruction reminders subroutine. 990 This POKE stops the computer asking 'scroll ?' at line 1070. 995 GOSUBs the traffic light countdown subroutine. Both this and the subroutine activated at line 950 occur at the start of each stage.
1000-1070	Contains the main body of Stage 1.
1900-1980	Checks for INKEY$ and detects collisions.
2000	Gives you a bonus for completing Stage 1 and GOSUBs the congratulations message (also activated each stage).
2020-2040	Determines the width of the roof and sets up the screen.
2100-2220	Contains the Stage 2 'action'.
2510-2590	Prints the chimney and determines the speed for Stage 3.
2600-2800	Stage 3 proper.
3010-3090	Initialises screen and variables for Stage 4.
3100-3130	Stage 4 routine when there is no snowball.
3200-3240	Ditto for when there is a snowball present.
3300-3330	Performs Santa's leap over the Christmas tree.
3600-3617	Sets up the variables and screen for the next stage. Note that the INK colour is set to 5 (cyan) yet the houses are printed in yellow so that their ATTRibutes are different. Thus a collision with the houses can be detected using the ATTR command in line 372.
3620-3800	Contains the main part of Stage 5.

From there on the REM statements serve just as well as a description would so you can go through them if you need a breakdown of the latter parts of the program — $ mainly subroutines.

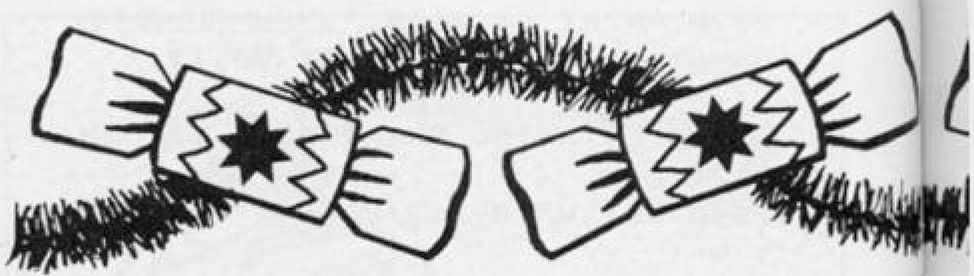
```
   1 REM ***********************
         *Underlined characters*
         *are entered in       *
         *GRAPHICS mode.       *
         ***********************
  10 REM INITIALISE SREEN AND
         SOME VARIABLES
  22 GO SUB 101
  25 LET H=0: DIM H$(10)
  30 LET Y=1: DIM K$(4): LET L=5
 : LET S=0
  40 BRIGHT 0: PAPER 0: BORDER 0
 : CLS : POKE 23658,8
  50 INK 6: PRINT AT 0,2; BRIGHT
 1;"SANTA'S CHRISTMAS NIGHTMARE!
";TAB 2;"____________________________
_____"
  52 GO SUB 9200
  55 REM DEFINE KEYS
  57 INK 6: PRINT AT 0,2; BRIGHT
 1;"SANTA'S CHRISTMAS NIGHTMARE!
";TAB 2;"____________________________
_____"
  60 INK 5: PRINT AT 5,0;"WHICH
KEYS WOULD YOU LIKE ?"''''"LEFT:"
''"RIGHT:"'''"UP:"'''"DOWN:"
  70 FOR F=1 TO 4: PRINT AT F*2+
6,10; FLASH 1;"?"
  80 LET R$=INKEY$: IF R$<>"" TH
EN  GO TO 80
  90 LET R$=INKEY$: IF R$="" THE
N  GO TO 90
  95 BEEP .05,20: PRINT AT F*2+6
,10;R$: LET K$(F)=R$: NEXT F
  97 FOR F=1 TO 4: FOR G=1 TO 4:
 IF F=G THEN  NEXT G: NEXT F: GO
 TO 100
  98 IF K$(F)=K$(G) THEN  CLS :
GO TO 57
  99 NEXT G: NEXT F
 100 GO TO 200
 101 REM DEFINE GRAPHICS
 103 RESTORE 100
 105 FOR G=144 TO 159: FOR F=0 T
O 7: READ A: POKE USR CHR$ G+F,A
: NEXT F: NEXT G
 110 DATA 28,28,8,62,8,28,20,20
 120 DATA 56,124,214,186,124,214
,146,16,24,52,82,255,169,255,153
,255,56,120,44,46,42,58,40,104,2
4,36,24,36,66,66,36,24
 130 DATA 18,63,254,252,124,62,1
27,118
 140 DATA 30,30,76,111,236,255,2
55,126,0,1,1,3,255,252,68,68
 150 DATA 1,3,7,15,31,63,127,255
,128,192,224,240,248,252,254,255
 160 DATA 102,102,102,102,255,25
5,255,255
 170 DATA 0,0,64,96,224,255,255,
126,28,62,127,127,127,73,42,28
 180 DATA 255,32,32,255,255,2,2,
255
 190 DATA 60,66,129,129,129,129,
66,60,60,126,255,255,255,255,126
,60
```

```
 199 RETURN
 900 REM SCREEN 1
 902 BRIGHT 0: PAPER 7: BORDER 7
: CLS
 905 INK 1: PRINT AT 11,10;"PLEA
SE WAIT"
 910 DIM P(Y*79,2): FOR F=1 TO Y
*79: LET P(F,1)=INT (RND*30)+1:
LET P(F,2)=INT (RND*3)+146: NEXT
 F
 920 INK 1: LET N=0: LET E=1: LE
T X=15
 950 GO SUB 9800: INK 2: CLS
 990 POKE 23692,255
 995 GO SUB 9900
1000 FOR F=1 TO 100: LET S=S+1
1050 PRINT AT 0,X;"A"
1060 IF F<80 THEN  FOR G=1 TO Y:
 LET N=N+1: PRINT AT 21,P(N,1);
INK 0;CHR$ P(N,2): NEXT G: PRINT
 AT 21,0; INK 4;"B";AT 21,31;"B"
1065 GO SUB 1900
1070 PRINT AT 21,0; INK 8; OVER
1,,,,: NEXT F: GO TO 2000
1900 LET R$=INKEY$: FOR G=1 TO 4
: IF K$(G)=R$ THEN  GO TO G*20+1
900
1910 NEXT G: GO TO 1960
1920 IF ATTR (1,X-1)<>58 THEN  P
RINT AT 1,X-1; INK 2;"E";AT 0,X;
" ": INK 1: GO SUB 9000: GO TO 9
00
1930 LET X=X-1: IF X=0 THEN  LET
 X=X+1: GO TO 1960
1935 RETURN
1940 IF ATTR (1,X+1)<>58 THEN  P
RINT AT 1,X+1; INK 2;"E";AT 0,X;
" ": INK 1: GO SUB 9000: GO TO 9
00
1950 LET X=X+1: IF X=31 THEN  LE
T X=X-1: GO TO 1960
1955 RETURN
1960 IF ATTR (1,X)<>58 THEN  PRI
NT AT 1,X; INK 2;"E";AT 0,X;" ":
 INK 1: GO SUB 9000: GO TO 900
1980 RETURN
2000 INK 1: LET S=S+50*L: GO SUB
 9820
2005 PAPER 0: BORDER 0: CLS
2010 REM STAGE 2
2020 INK 5: LET E=2: GO SUB 9800
: INK 6: LET W=10-Y: IF W<1 THEN
  LET W=1
2030 LET C$="[illegible]"( TO W)
2040 PRINT AT 15,16;"K";TAB 16;"
■"
2100 PRINT AT 17,15-W/2;"I";C$;"
2": FOR F=18 TO 21: PRINT AT F,1
5-W/2;C$;"■": NEXT F
2110 GO SUB 9900: FOR F=0 TO 29:
 PRINT AT 0,F; INK 2;" G"; INK 6
;"H": IF INKEY$=K$(4) THEN  GO T
O 2200
2120 NEXT F: INK 5: GO SUB 9000:
 GO TO 2010
2200 LET X=F+1: PRINT AT 0,X; IN
K 2;"L"; INK 6;"H": INK 2: FOR F
=2 TO 13: PRINT AT F-1,X;" ";AT
F,X;"M";AT F+1,X;"A": FOR G=1 TO
 40: NEXT G: NEXT F: IF X=16 THE
N  LET S=S+100: GO TO 2250
2210 FOR F=14 TO 15: PRINT AT F-
1,X;" ";AT F,X;"M";AT F+1,X;"A":
 FOR G=1 TO 40: NEXT G: NEXT F
2220 IF X<16-(W/2+1) OR X>16+W/2
 THEN  FOR F=16 TO 20: PRINT AT
F-1,X;" ";AT F,X;"M";AT F+1,X;"A
": FOR G=1 TO 40: NEXT G: NEXT F
: INK 5: GO SUB 9000: GO TO 2010
2250 CLS
2300 INK 5: LET S=S+50+20*L: GO
SUB 9820
2500 REM STAGE 3
2510 CLS : LET E=3: INK 5: GO SU
B 9800: INK 6: FOR F=5 TO 21: PR
INT AT F,12;"N";TAB 19;"N": NEXT
 F: PRINT AT 4,12;"NNNNNNNN": FO
R F=0 TO 3: PRINT AT F,13;"NN";T
AB 17;"NN": NEXT F
```

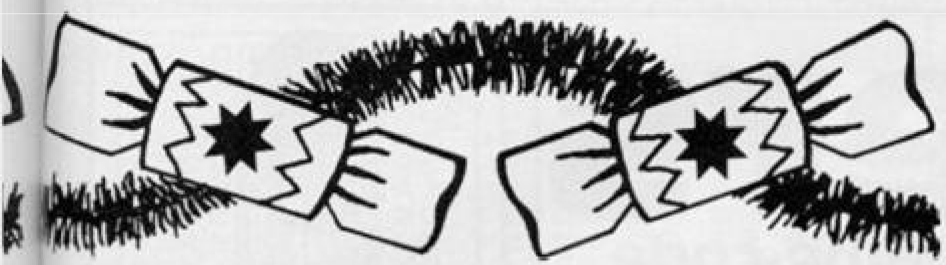


```
2520 PRINT AT 21,Ø;"NNNNNNNNNNNNNNNN";AT 6,13;"NNN";AT 2Ø,12;"N"
2530 LET X=15: PRINT AT 5,15; INK 2;"A"
2540 LET W=1Ø-Y: IF W<1 THEN LET W=1
2550 LET W=W/2
2590 GO SUB 9900
2600 INK 5: FOR F=21 TO 6 STEP -1: FOR G=1 TO W*4: NEXT G: PRINT AT F,16;"■■■": IF F<21 THEN PRINT AT F+1,16;"   "
2610 LET R$=INKEY$: IF R$=K$(2) THEN GO TO 2700
2615 FOR G=1 TO W*4: NEXT G: PRINT AT F,16;"■■■": IF INKEY$=K$(2) THEN GO TO 2700
2620 NEXT F: FOR F=6 TO 21: FOR G=1 TO W*4: NEXT G: PRINT AT F-1,16;"   ";AT F,16;"■■■": IF INKEY$=K$(2) THEN GO TO 2700
2625 FOR G=1 TO W*4: NEXT G: IF F<21 THEN PRINT AT F,16;"■■■"
2627 IF INKEY$=K$(2) THEN GO TO 2700
2630 NEXT F
2640 PRINT AT 17,Ø;"TIME:": PRINT AT 2Ø,12;"    ": FOR T=2Ø TO Ø STEP -1: PRINT AT 17,5;T;" ": NEXT T: FOR F=1 TO 4Ø: NEXT F: INK 5: GO SUB 9ØØØ: GO TO 25ØØ
2650 FOR F=6 TO 21: FOR G=1 TO W*4: NEXT G: PRINT AT F,16;"■■■"; AT F-1,16; INK 2;"A": IF INKEY$=K$(1) THEN GO TO 269Ø
2658 PRINT AT F-1,16; INK 2;"A ": FOR G=1 TO W*4: NEXT G: IF F<21 THEN PRINT AT F,16;"■■■"
2659 IF INKEY$=K$(1) THEN GO TO 269Ø
2660 PRINT AT F-1,16;"   ": NEXT F: PRINT AT F-2,16; INK 2;"A": PRINT AT 17,Ø;"TIME:": PRINT AT 2Ø,14;" ": FOR T=2Ø TO Ø STEP -1: PRINT AT 17,5;T;" ": IF INKEY$=K$(1) THEN GO TO 28ØØ
2670 NEXT T: FOR F=1 TO 4Ø: NEXT F: INK 5: GO SUB 9ØØØ
2690 PRINT AT F-1,16;"    ";AT F,16;"■■■": FOR F=F-1 TO 2Ø: PRINT AT F-1,15;" ";AT F,15; INK 2;"A": FOR G=1 TO 4Ø: NEXT G: NEXT F: PRINT AT 2Ø,15; INK 2;"E": FOR F=1 TO 4Ø: NEXT F: INK 5: GO SUB 9ØØØ: GO TO 25ØØ
2700 LET X=16: PRINT AT 5,15;" ": IF F=6 THEN GO TO 265Ø
2710 IF F>6 THEN FOR F=6 TO F-1: PRINT AT F-1,16;" ";AT F,16; INK 2;"A": FOR G=1 TO 4Ø: NEXT G: NEXT F: PRINT AT F-1,16; INK 2;"E": FOR F=1 TO 4Ø: NEXT F: INK 5: GO SUB 9ØØØ: GO TO 25ØØ
2800 FOR F=15 TO Ø STEP -1: PRINT AT 2Ø,F; INK 2;"A ": FOR G=1 TO 2Ø: NEXT G: NEXT F
2900 INK 5: LET S=S+5Ø+(T+1Ø)*L: GO SUB 982Ø
3000 REM STAGE 4
3010 LET E=4: INK 5: GO SUB 98ØØ
3020 INK 4: FOR F=1Ø1 TO 61 STEP -2Ø: FOR G=1 TO 3Ø: PLOT 3Ø-G/2,F-G: DRAW G,Ø: NEXT G: NEXT F: INK 2: FOR F=26 TO 34: PLOT F,16: DRAW Ø,15: NEXT F
3030 INK 5: FOR F=Ø TO 15: PLOT 2Ø-F/2,F: DRAW F+2Ø,Ø: NEXT F
3040 LET W=1Ø-Y: LET W=W/3: IF W<1 THEN LET W=1
3090 LET X=3Ø: INK 7: GO SUB 99ØØ
3100 PRINT AT 21,X; INK 2;"A "
3110 FOR F=1 TO W*4: NEXT F: IF INKEY$=K$(1) THEN LET X=X-1: LET S=S+1: IF X=6 THEN GO TO 33ØØ
3120 IF RND<.15 AND X>11 THEN LET U=Ø: PRINT AT 21,X; INK 2;"A": GO TO 32ØØ
3130 GO TO 31ØØ
3200 FOR F=7 TO 31: FOR G=1 TO W*4: NEXT G: LET S=S+1: PRINT AT 21,F-1;" P": IF F=X AND U=Ø THEN PRINT AT 21,X; INK 2;"E": FOR F=1 TO 2Ø: NEXT F: INK 5: GO SUB 9ØØØ: GO TO 3ØØØ
3210 IF INKEY$=K$(3) AND U=Ø THEN LET U=1: LET T=Ø: PRINT AT 21,X;" "
3220 IF U=1 THEN LET T=T+1: IF T=5 THEN LET U=Ø: PRINT AT 2Ø,X;" "
3230 PRINT AT 21-U,X; INK 2;"A": IF F=X AND U=Ø THEN PRINT AT 2
```





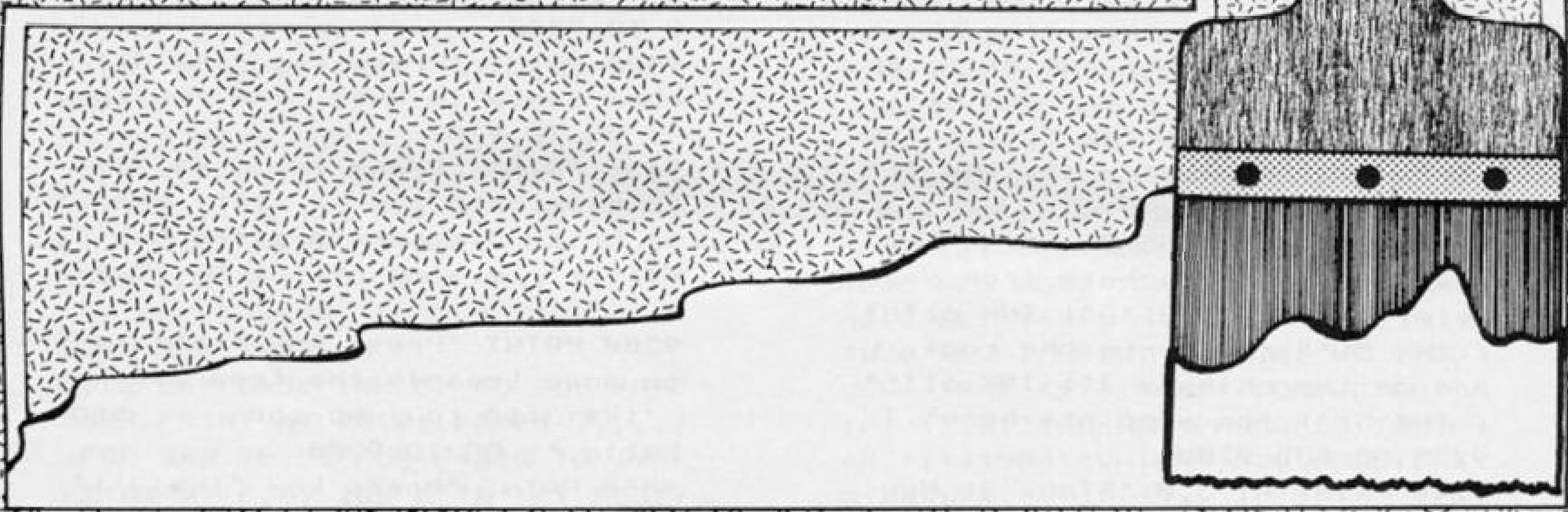
```
1,X; INK 2;"E": FOR F=1 TO 20: N
EXT F: INK 5: GO SUB 9000: GO TO
 3000
3240 NEXT F: LET U=0: PRINT AT 2
1,31;" ";AT 21,X; INK 2;"A";AT 2
0,X;" ": LET S=S+5: GO TO 3100
3300 PRINT AT 21,7;" ": FOR F=20
 TO 10 STEP -1: PRINT AT F,6; IN
K 2;"A";AT F+1,6;" ": FOR G=1 TO
 10: NEXT G: NEXT F
3310 PRINT AT 10,6;" ": RESTORE
3320: FOR F=1 TO 5: READ A,B: PR
INT AT A,B; INK 2;"A": FOR G=1 T
O 10: NEXT G: PRINT AT A,B;" ":
NEXT F
3320 DATA 9,5,8,4,7,3,8,2,9,1
3330 FOR F=10 TO 21: PRINT AT F,
0; INK 2;"A";AT F-1,0;" ": FOR G
=1 TO 10: NEXT G: NEXT F
3340 FOR F=1 TO 200: NEXT F: INK
 5: LET S=S+50: GO SUB 9820
3500 REM STAGE 5
3510 LET E=5: INK 5: GO SUB 9800
3600 PRINT AT 20,0; PAPER 7,,,,:
 FOR G=0 TO 5: FOR F=6+G TO 19:
PRINT AT F,G; INK 6;"■";AT F,31-
G;"■": NEXT F: PRINT AT G+5,G; I
NK 6;"J";AT G+5,31-G;"I": NEXT G
: PRINT AT 6,2; INK 6;"K";AT 7,2
;"■";AT 6,29;"K";AT 7,29;"■"
3610 PRINT AT 19,15; INK 2;"G";
INK 6;"H": LET T=0
3615 GO SUB 9900: IF INKEY$=K$(3
) THEN  PRINT AT 11,11;"HANDS OF
F!": FOR F=1 TO 100: NEXT F: PRI
NT AT 11,11;"          ": GO TO
3615
3617 PRINT AT 2,2; INK 7;"TIME:5
00"
3620 FOR F=500 TO 0 STEP -1
3625 PRINT AT 2,7; INK 7;F;" "
3630 IF INKEY$=K$(3) THEN  LET T
=T+1: GO TO 3650
3640 NEXT F: GO TO 3700
3650 IF INKEY$<>"" THEN  LET F=F
-1: PRINT AT 2,7; INK 7;F;" ": I
F F>0 THEN  GO TO 3650
3655 IF F<1 THEN  GO TO 3700
3660 NEXT F
3700 LET T=T*2: LET T=T-15*Y: LE
T T=T-36: LET T=19-T: IF T<0 THE
N  LET T=0
3710 FOR F=18 TO T STEP -1: PRIN
T AT F,15; INK 2;"G"; INK 6;"H";
AT F+1,15;"  ": FOR G=1 TO 40: N
EXT G: LET S=S+20: NEXT F
3720 FOR F=15 TO 29: FOR G=1 TO
40: NEXT G: PRINT AT T,F; INK 2;
" G"; INK 6;"H": IF ATTR (T,F+3)
=6 THEN  FOR G=1 TO 40: NEXT G:
GO TO 3750
3730 NEXT F: GO TO 3800
3750 INK 2: PRINT AT T,F+1;" G":
 FOR G=1 TO 40: NEXT G: PRINT AT
 T,F+2;"A": FOR J=1 TO 40: NEXT
J: PRINT AT 19,F+2;"E": INK 5: G
O SUB 9000: GO TO 3500
3800 LET S=S+200: GO SUB 9820
4000 REM NEXT YEAR
4010 LET Y=Y+1
4020 CLS : PRINT AT 6,0;"WELL DO
NE - YEAR ";Y+1983;" COMPLETED!"
;AT 12,0;"YOU NOW PROGRESS TO YE
AR ";Y+1984'"WHICH, OBVIOUSLY, I
S HARDER."
4030 FOR F=1 TO 200: NEXT F: GO
TO 900
9000 REM YOU LOSE A LIFE
9010 LET L=L-1: FOR F=1 TO 100:
NEXT F: CLS : PRINT AT 6,2;"BAD
LUCK ! - YOU LOST A LIFE";AT 12,
10;L;" LIVES LEFT"
9015 FOR F=15 TO 0 STEP -1: BEEP
 .1,F: NEXT F
9020 FOR F=1 TO 200: NEXT F: CLS
 : IF L=0 THEN  GO TO 9100
9030 INK 2: RETURN
9100 PRINT AT 4,0;"YOU REACHED Y
EAR ";Y+1984;" (STAGE ";E;")";AT
 8,0;"YOU SCORED ";S
9110 IF S>=H THEN  PRINT AT 12,0
;"YOU BEAT THE HIGH SCORE
  THE HIGH SCORE IS NOW ";S: LET
 H=S: PRINT "PLEASE ENTER YOUR N
AME": INPUT  LINE H$
9120 IF S<H THEN  PRINT AT 12,0;
"THE HIGH SCORE REMAINS ";H'"BY
";H$
9130 PRINT AT 19,0;"PRESS ANY KE
Y TO PLAY AGAIN"
9140 IF INKEY$="" THEN  GO TO 91
40
9150 RESTORE : CLS : GO TO 30
9210 BRIGHT 1: INK 5: PRINT ''"Yo
u play the part of Santa ("; INK
 2;"A"; INK 5;")"'"delivering pr
```



```
esents to the last house on his
round."
9220 PRINT '"STAGE 1: Use the le
ft and right keys to guide Santa
 through the snow avoiding the t
rees ("; INK 4;"B"; INK 5;"), th
eigloos ("; INK 7;"C"; INK 5;"),
 the penguins ("; INK 7;"D"; INK
 5;") andthe snowmen ("; INK 7;"
E"; INK 5;")."
9230 PRINT '"STAGE 2: Press the
down key to  parachute from your
 sleigh ("; INK 2;"G"; INK 6;"H"
; INK 5;") down onto the roof. L
and on the chimney ("; INK 6;"K"
; INK 5;") for a points bonus."
9235 GO SUB 9700
9240 PRINT AT 3,0;"STAGE 3: Now
you are inside the chimney and m
ust get to the      bottom using
the lift (▬▬).      Press the rig
ht key to step ontothe lift and,
 when the brick ("; INK 6;"N"; I
NK 5;")at the bottom disappears,
 press the right key before your
 time  limit runs out."
9250 PRINT '"STAGE 4: Cross the
floor to the Christmas tree usin
g the left   key. When a snowbal
l ("; INK 7;"P"; INK 5;") appear
swait until it is near you and
 use the up key to jump over it.
 Wait until the snowball leaves
 the screen and then continue
 towards the Christmas tree."
9255 GO SUB 9700
9260 PRINT AT 3,0;"STAGE 5: Pres
s the up key as    many times as
 you can while the clock ticks a
way. The more timesyou press it
the more height thesleigh ("; IN
K 2;"G"; INK 6;"H"; INK 5;") wil
l achieve. You    must get high e
nough to clear    the houses othe
rwise you'll      crash."
9270 PRINT '"At the beginning of
 each screen you will be given a
 brief           reminder of these i
nstructions  and then traffic li
ghts will    count you down to t
he start.    When the green ligh
t ("; INK 4;"O"; INK 5;") lights
 up ("; INK 4;"P"; INK 5;") the
screen will start."
9280 GO TO 9700
9300 PRINT AT 8,12;"REMEMBER": P
RINT AT 9,0: GO TO E*50+9300
9350 PRINT "Use keys ";K$(1);" a
nd ";K$(2);" to avoid the   obst
acles. Be prepared for the  spee
d increase towards the end !": G
O TO 9700
9400 PRINT "Press key ";K$(4);"
when Santa is over  the rooftop.
": GO TO 9700
9450 PRINT "Press key ";K$(2);"
to step onto the     lift. When t
he brick disappears press key ";
K$(1);" to step off.": GO TO 970
0
9500 PRINT "Press key ";K$(1);"
to move towards the tree and key
 ";K$(3);" to jump the      snow
balls.": GO TO 9700
9550 PRINT "Press key ";K$(3);"
as fast as you can  to achieve h
eight. Don't hold   key ";K$(3);
" down, it won't work !"
9700 PRINT AT 21,3;"PRESS ANY KE
Y TO CONTINUE"
9710 IF INKEY$="" THEN  GO TO 97
10
9720 PRINT AT 3,0,,,,,,,,,,,,,,,
,,,,,,,,,,,,,,,,,,,,,,,
9730 RETURN
9800 REM NEXT SCREEN MESSAGE
9810 CLS : PRINT AT 6,11;"YEAR:"
;Y+1984;AT 12,12;"STAGE ";E: FOR
 F=1 TO 80: NEXT F: CLS : GO SUB
 9300: RETURN
9820 REM END OF SCREEN MESSAGE
9830 CLS : PRINT AT 5,11;"YEAR:"
;Y+1984;AT 10,0;"WELL DONE ! - S
TAGE ";E;" COMPLETED";AT 15,11;"
SCORE:";S
9840 FOR F=10 TO 25: BEEP .1,F:
NEXT F: FOR F=1 TO 200: NEXT F
9850 RETURN
9900 REM TRAFFIC LIGHTS
9910 PRINT AT 1,0; INK 2;"P";AT
2,0; INK 6;"O";AT 3,0; INK 4;"O"
: FOR F=1 TO 50: NEXT F: PRINT A
T 2,0; INK 6;"P": FOR F=1 TO 50:
 NEXT F: PRINT AT 1,0; INK 2;"O"
;AT 2,0; INK 6;"O";AT 3,0; INK 4
;"P": RETURN
```

Painters

A brilliant two-player machine-code game which is a 1K masterpiece from Barry Smith.



Once this program has been entered then SAVE it before RUNning as the slightest error will cause all your work to be lost!

The plot

Two decorators are frantically rushing around a room, each desperately trying to paint as much of the floor as possible. You must guide their brushes to help them achieve this aim. Ten points are awarded for each brush stroke that covers an unpainted section of floor. If, in their haste, either worker strays onto those parts of the room that have so far been painted they will be fired, this news causes them to flush angrily. The last worker to be sacked is the one to whom the score is given. Tea breaks are allowed only at the discretion of either player.

Control keys

Painter One, who starts work at the left of the room is controlled with the following keys:

"Shift" — Up
"Z" — Down
"X" — Left
"C" — Right

Painter two, who starts work at the right of the room is controlled by a second set of keys:

"N" — Up
"M" — Down
"." — Left
"Space" — Right

Press "T" and both workers stop for a tea break, until you tell them to start work again by pressing "Y".

Entering the game

Type in Line 1 of the Hex Loader, it has 321 characters after the REM statement, that is 10 full lines and one extra character. Now, as a direct command type CLS, then enter Line 2 which has four full lines and 19 extra characters after its REM. You may now enter the rest of the program normally and need not worry if any of the lines float to the top of the screen. RUN the program. ENTER one line of code from the Hex Dumps. After the last line, from Hex Dump 2, has been typed in, the computer will stop with the report 0/80. Delete lines 80-10 in that order and add the line 3 RAND USR 16536. the fame is now ready for playing.
GOOD LUCK.

Pokes

You may like to POKE:
16537 — the paint used by Painter one.
16542 — the paint used by Painter Two.
16828 — the speed of the game. This is initially 13 and ranges from, 1 = impossibly fast, to 255 = incredibly slow.

```
HEX LOADER
--- ------

   1 REM EEEEEEEEEEEEEEEEEEEEEEE
EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE
EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE
EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE
EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE
EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE
EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE
EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE
EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE
EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE
EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE
EEEEEEEEEE
   2 REM EEEEEEEEEEEEEEEEEEEEEEE
EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE
EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE
EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE
EEEEEEEEEEEEEEEEEEEEEEEEEEEE
  10 LET A$=""
  20 FOR A=16514 TO 16987
  30 LET A=A+(A=16835)*6
  40 IF A$="" THEN INPUT A$
  50 POKE A,16*CODE A$+CODE A$(2
) -476
  60 PRINT AT 0,0;A;" ";A$( TO 2
)
  70 LET A$=A$(3 TO )
  80 NEXT A
```

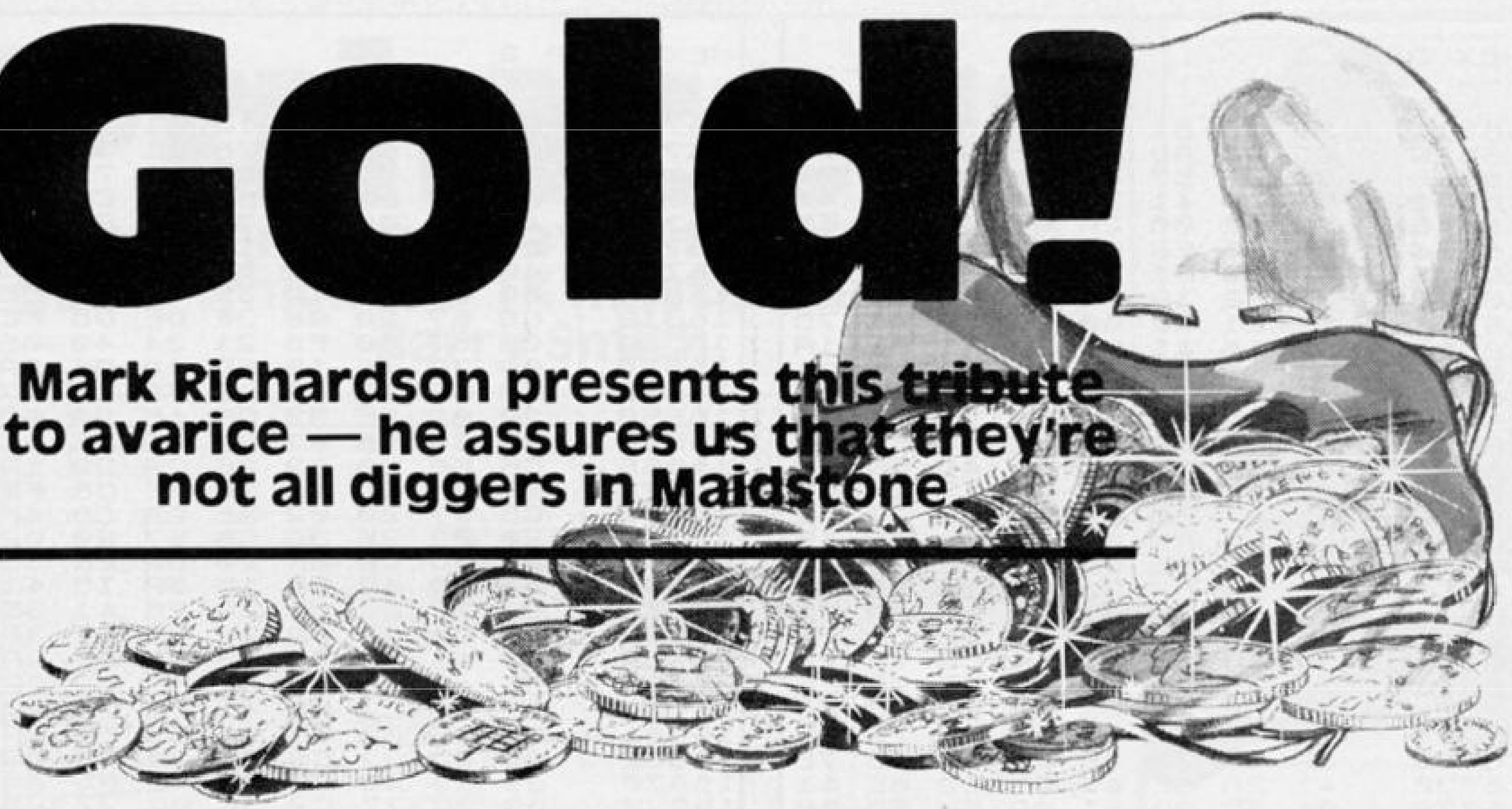
HEX DUMP 1

16514	:	B8	A8	B4	B7	AA	08	9C	9C
16522	:	9C	9C	AD	AE	08	9C	9C	9C
16530	:	9C	A7	B7	BA	B8	AD	3E	97
16538	:	32	2C	41	3E	97	32	44	41
16546	:	01	6E	00	2A	0C	40	E5	E5
16554	:	E5	E5	E5	09	22	26	41	01
16562	:	85	00	E1	09	22	3E	41	3E
16570	:	2A	32	1A	42	3E	3C	32	2D
16578	:	42	21	00	00	22	7E	41	7C
16586	:	32	18	41	06	0B	E1	0E	1A
16594	:	23	C5	E5	CD	9B	09	E1	C1
16602	:	11	08	80	72	79	FE	1A	20
16610	:	01	73	0D	20	EB	73	23	10
16618	:	E5	06	19	2B	73	10	FC	0E
16626	:	0A	D1	13	21	82	40	ED	B0
16634	:	3E	08	06	09	12	13	10	FC
16642	:	0E	07	ED	B0	01	79	00	11
16650	:	9D	A6	E1	09	01	34	40	15
16658	:	72	3E	31	02	0A	FE	00	20
16666	:	FB	7A	BB	20	F2	16	80	72
16674	:	21	18	41	01	00	00	0A	FE
16682	:	80	3E	97	02	28	0D	CB	FE
16690	:	C6	80	32	2C	41	CB	76	20
16698	:	02	CB	EE	01	00	00	0A	FE
16706	:	80	3E	97	02	28	07	CB	F6
16714	:	C6	80	32	44	41	CB	7E	28
16722	:	2A	CB	76	28	26	ED	4B	7E
16730	:	41	2A	8E	41	09	22	8E	41
16738	:	01	0C	00	E1	09	54	5D	0E
16746	:	05	21						

HEX DUMP 2

16748	:	93	40	ED	B0	13	21	18	41
16756	:	3E	9D	CB	6E	28	01	3C	12
16764	:	C9	21	00	00	23	22	7E	41
16772	:	01	09	00	E1	E5	09	CD	4E
16780	:	42	21	00	00	7C	B5	28	06
16788	:	2B	22	8E	41	18	0E	21	91
16796	:	40	CD	4E	42	0E	19	E1	E5
16804	:	09	CD	4E	42	3E	FB	DB	FE
16812	:	CB	67	20	08	3E	DF	DB	FE
16820	:	CB	67	20	F8	21	34	40	06
16828	:	0D	70	46	10	FD	18	06	2A
16842	:	1A	42	3E	FE	DB	FE	CB	47
16850	:	20	02	2E	33	CB	4F	20	02
16858	:	2E	38	CB	57	20	02	2E	3C
16866	:	CB	5F	20	02	2E	2A	22	1A
16874	:	42	2A	2D	42	3E	7F	DB	FE
16882	:	CB	47	20	02	2E	2A	CB	4F
16890	:	20	02	2E	3C	CB	57	20	02
16898	:	2E	38	CB	5F	20	02	2E	33
16906	:	22	2D	42	0E	1B	3A	18	41
16914	:	CB	7F	20	0B	2A	26	41	3E
16922	:	2A	CD	37	42	22	26	41	3A
16930	:	18	41	CB	77	C2	22	41	2A
16938	:	3E	41	3E	3C	CD	37	42	22
16946	:	3E	41	C3	22	41	FE	33	20
16954	:	05	C6	00	ED	42	C9	FE	2A
16962	:	20	02	23	C9	FE	38	20	02
16970	:	09	C9	2B	C9	7E	FE	A5	28
16978	:	03	3C	77	C9	3E	9C	77	2B
16986	:	18	F2						

Gold!



Mark Richardson presents this tribute to avarice — he assures us that they're not all diggers in Maidstone.

The lure of that yellow stuff, I can almost smell it, more than a teacher gets in a week! The trouble is that nothing comes easy and the previous owner put some pretty formidable obstacles in the way of getting hold of it.

I know it's in the castle but the first job is getting in there, the gatekeeper wants a passnumber before letting me enter and I think he has cannabalistic tendencies. Still, he's pretty thick and keeps saying if I'm too high or low. Once inside there are rumoured to be more varied and objectionable nasties to overcome before finding the gold and escaping with it.

Mark has produced an entertaining program which shows how a few interesting, but in themselves uninspired, games can be linked together to produce a much more effective game. Random factors have been used with the aim of producing a game which makes it a challenge to everyone — including the programmer who types it in, as there are no preset codes which will help the typist solve the game.

To enhance the end result a variety of BEEPs and graphics have been used at appropriate points. As the program is essentially a suite of subgames, variables are repeated in each section and kept to a minimum and are:

A	Line 520, centre line of gateway.
B/C	Line 4210, points of sword.
D	Man.
E	Bag of gold.
F	Hound.

The POKE 23658,8 is to set Caps Lock on, so that all entries are made in capitals and saves double checking for upper/lower case.

```
   1 REM **********************
         *Underlined characters*
         *are entered in       *
         *GRAPHICS mode.       *
         **********************

   5 REM ########GOLD!########
  13 REM
  15 CLS : PAPER 7: CLS
  20 FOR N=1 TO 140: PRINT  INK
6;"      ";: NEXT N: BORDER 6
  30 PRINT AT 21,29; INK 6;"
"
  40 FOR M=8 TO 12: PRINT AT M,1
0; PAPER 6; BRIGHT 1;"
  ": NEXT M
  50 GO SUB 8600
  60 FOR N=1 TO 5: FOR M=20 TO 4
0: BEEP .02,M: NEXT M: NEXT N
  70 BORDER 6: INK 0: PAPER 7: C
LS
  80 GO SUB 8000
  90 POKE 23658,8
  99 REM ###INSTRUCTIONS###
 100 CLS
 110 PRINT "YOU ARE SEARCHING FO
R "; INK 0; BRIGHT 1; PAPER 6;"G
OLD"
 120 PRINT : PRINT : PRINT : PRI
NT
 130 PRINT "BEHIND THE CASTLE GA
TE IS A      FIERCE GATE-KEEPER."
: PRINT " HE HAS ASKED YOU FOR T
HE               PASSNUMBER! ": P
RINT " IT IS BETWEEN 1 AND 100.
      HE WILL GIVE YOU 6 GUESSE
S.": PRINT " BE CAREFUL THOUGH H
E'S "; INK 2;"VERY "; INK 0;"
 HUNGRY!"
 140 PRINT : PRINT  INK 3;"PRESS
 ANY KEY TO CONTINUE": PAUSE 0:
CLS
```



```
499 REM ###PRINT    GATE###
500 INK 2
510 PRINT AT 7,9;"
 "
520 LET Q$="
 "
530 FOR N=8 TO 13
540 PRINT AT N,8;Q$: NEXT N
550 PRINT AT 14,8;"    O O
 "
560 FOR N=15 TO 19
570 PRINT AT N,8;Q$: NEXT N
580 FOR M=10 TO 40 STEP 2: BEEP
.02,M: NEXT M
590 INK 0
600 LET TR=6: LET X=1+INT (RND*
100)
999 REM ###GUESS    NUMBER###
1000 IF TR=0 THEN  CLS : PRINT A
T 10,0;"SORRY YOU HAVE BEEN EATE
N BY THE GATE KEEPER-YOU ARE DEA
D": FOR N=1 TO 3: BEEP 1,-5: BEE
P 1.5,-10: NEXT N: GO TO 8500
1010 INPUT "PICK A NUMBER BETWEE
N 1-100 ";Z
1020 IF Z<1 OR Z>100 THEN  GO TO
 1010
1030 IF X=Z THEN  BORDER 0: PAPE
R 1: CLS : INK 6: PRINT AT 0,0;"
YOU ARE AMAZING,YOU HAVE FOUND
THE  PASSNUMBER": FOR N=1 TO 50:
 BEEP .01,N: NEXT N: FOR N=50 TO
 1 STEP -1: BEEP .01,N: NEXT N:
BEEP 1,25: GO TO 1500
1040 IF Z>X THEN  PRINT AT 0,0;"
TOO HIGH,TRY AGAIN": LET TR=TR-1
: GO TO 1000
1050 IF Z<X THEN  PRINT AT 0,0;"
TOO LOW,TRY AGAIN  ": LET TR=TR-
1: GO TO 1000
1499 REM ###THREE   DOORS###
1500 BORDER 0: PAPER 1: INK 6: C
LS
1505 PAUSE 50: PLOT 90,0: DRAW 0
,100: DRAW -90,10
1510 PLOT 165,0: DRAW 0,100: DRA
W 90,10
1520 CIRCLE 127,80,10: PLOT 127,
70: DRAW 0,-20: DRAW 15,-20
1530 PLOT 112,30: DRAW 15,20: PL
OT 112,55: DRAW 15,15: DRAW 15,-
15
1540 LET W$="       "
1550 LET X$="   O "
1560 FOR N=0 TO 1: PRINT AT N,1;
 INK 4;W$;AT N,13; INK 2;W$;AT N
,25; INK 6;W$: NEXT N
1570 PRINT AT 2,1; INK 4;X$;AT 2
,13; INK 2;X$;AT 2,25; INK 6;X$
1580 FOR N=3 TO 5
1590 PRINT AT N,1; INK 4;W$;AT N
,13; INK 2;W$;AT N,25; INK 6;W$:
 NEXT N
1600 PRINT AT 0,0;"1";AT 0,12;"2
";AT 0,24;"3"
1700 LET DR=1+INT (RND*3)
1710 INPUT "WHICH DOOR?";CD
1720 IF CD=DR THEN  GO TO 2000
```

```
1730 IF CD<>DR THEN  GO TO 4000
1999  REM ### GOLDSEARCH ###
2000 BORDER 5: PAPER 7: INK 0: C
LS
2010 PRINT AT 8,0;"YOU'VE FOUND
THE RIGHT DOOR      THE GOLD IS H
IDDEN IN THIS ROOM.TO FIND IT YO
U MUST MOVE WITH    THE CURSOR KE
YS.  GOOD LUCK!"
2020 FOR N=1 TO 6: FOR M=0 TO 40
 STEP 5: BEEP .15,M: NEXT M: NEX
T N: CLS
2030 LET X=INT (RND*22)
2035 LET Y=INT (RND*32)
2040 PRINT AT X,Y; INK 7;"G"
2050 LET A=12: LET B=12
2060 PRINT AT A,B;" "
2070 IF INKEY$="" THEN  GO TO 20
70
2075 IF INKEY$="5" THEN  PRINT A
T A,B;" ": LET B=B-1
2080 IF INKEY$="6" THEN  PRINT A
T A,B;" ": LET A=A+1
2085 IF INKEY$="7" THEN  PRINT A
T A,B;" ": LET A=A-1
2090 IF INKEY$="8" THEN  PRINT A
T A,B;" ": LET B=B+1
2100 IF A<=0 THEN  LET A=0
2105 IF A>=21 THEN  LET A=21
2110 IF B<=0 THEN  LET B=0
2115 IF B>=31 THEN  LET B=31
2120 IF ATTR (A,B)=63 THEN  GO T
O 2500
2130 GO TO 2060
2200 PRINT AT A,B;" "
2499 REM ###FIND   GOLD###
2500 BORDER 2: PAPER 6: INK 2: C
LS
2505 PRINT  FLASH 1;AT 10,0;"!!!
!YOU HAVE FOUND THE GOLD!!!!!":
FOR N=1 TO 10: BEEP .2,25: BEEP
.1,15: BEEP .3,20: NEXT N
2508 PRINT AT A,B;" ": PAUSE 15
0
2510 BORDER 6: PAPER 7: CLS : IN
K 0
2520 PRINT AT 4,1;"YOU STILL HAV
E A DREADFUL        HOUND TO OVER
COME BEFORE YOU     CAN TAKE YOUR
 GOLD HOME."
2530  PRINT AT 8 ,1;"HE IS GUARD
ING THE EXIT TO THE  CASTLE.  LU
CKILY YOU FIND THREE STONES ON T
HE GROUND.             CAN YOU KIL
L HIM WITH ONE OF    THESE?"
2540 PRINT AT 14,1;"CHOOSE YOUR
ANGLE CAREFULLY      (Between 50
and 80 degrees)"
2550 PAUSE 800
2599  REM ## CONQUER HOUND ###
2600 BORDER 4: PAPER 1: INK 6: C
LS
2610 LET H=INT (RND*25+5)
2620 PRINT AT 21,0; INK 7;" ";A
T 21,1; INK 2;"

          "
2630 PRINT  INK 6;AT 19,H;"
";AT 20,H;"    ";AT 21,H;"
";AT 21,H; BRIGHT 1;" "
2640 FOR S=1 TO 3
2650 PRINT AT 1,0;"SHOT NO.";S:
INPUT "Enter angle of shot"; A
2660 LET A=INT A: IF A>80 THEN
INPUT "Angle too great,enter aga
in";A
2665 IF A<50 THEN  INPUT "Angle
too small, enter again";A
2670 FOR N=0 TO 100
2680 LET X=INT (0.062*N*(90-A)):
 LET B=N*(100-N): LET Y=INT (B*0
.05): PLOT X,Y
2690 BEEP 0.1,B/100
2700 LET L=21-INT (Y/8): LET C=I
NT (X/8): IF ATTR (L,C)>64 THEN
 GO TO 2750
2710 NEXT N
2750 IF C=H THEN  GO TO 2800
2780 NEXT S: IF S>3 THEN  PRINT
AT 10,2;"OH, DEAR, YOU FAILED.
          NOW YOU WILL STARVE TO
DEATH     IN THE CASTLE": PAUSE 2
50: GO TO 8500
2790 INPUT "Press ENTER for anot
her shot";Z$: GO TO 2640
2800  PRINT  FLASH 1;"YOU HAVE K
ILLED THE DREADFUL      HOUND. NO
W YOU MAY GO THROUGH  THE DOOR W
ITH THE GOLD": FOR N=1 TO 5: BEE
P .5,12: BEEP .75,20: NEXT N: GO
 TO 3000
2999 REM ###GET    HOME###
3000 BORDER 4: PAPER 7: CLS : IN
K 0
3010 PRINT AT 5,0;" YOU ESCAPE F
ROM THE CASTLE AND  FIND YOURSEL
F WITH THE GOLD AT    THE TOP OF
A GRASSY HILL."
3020 PAUSE 100
3030 IF RND>.7 THEN  PRINT AT 10
,0;"SUDDENLY YOUR BAG OF GOLD
     DROPS, AND IN YOUR HASTE TO
     RETRIEVE IT YOU TRIP.
       YOU ARE FALLING....": FOR N
```

```
=14 TO 21: PRINT AT N,3;"falling
...": NEXT N: FOR N=1 TO 3: FOR
M=5Ø TO Ø STEP -2: BEEP .Ø3,M: N
EXT M: NEXT N: BEEP 1.5,-1Ø: BEE
P 2,-25: GO TO 85ØØ
3Ø4Ø PRINT AT 1Ø,Ø;"YOU WALK DOW
N THE HILL SLOWLY.  YOU ARRIVE H
OME AND LIVE IN      PEACE WITH A
LL YOU NEED."''"      CONGRATULAT
IONS!": FOR B=5 TO 35 STEP 5: BE
EP .5,B: NEXT B: FOR B=36 TO 4 S
TEP -4: BEEP .3,B: NEXT B
3Ø5Ø PAUSE 1ØØ: GO TO 85ØØ
3999 REM ###LIFT###
4ØØØ LET M=1+INT (RND*2)
4Ø1Ø IF CD=M THEN  BORDER 6: INK
 Ø: PAPER 7: CLS : LET TR=6: LET
 X=1+INT (RND*1ØØ): PRINT AT 1Ø,
1;"IT WAS A LIFT!
   LOOK WHERE YOU ARE NOW!!
   (PRESS ANY KEY TO SEE)": PAUS
E Ø: CLS : GO TO 5ØØ
4Ø19 REM ###DRAGON###
4Ø2Ø IF CD<>M THEN  BORDER 2: PA
PER 7: CLS : INK Ø: PRINT  INK 1
; PAPER 6;"THERE IS A DRAGON!!!"
4Ø3Ø DIM A$(1Ø,18)
4Ø4Ø LET A$(1)="
       "
4Ø45 LET A$(2)="
       "
4Ø5Ø LET A$(3)="
        "
4Ø55 LET A$(4)="
       "
4Ø6Ø LET A$(5)="
       "
4Ø65 LET A$(6)="
       "
4Ø7Ø LET A$(7)="
        "
4Ø75 LET A$(8)="
        "
4Ø8Ø LET A$(9)="
       "
4Ø85 LET A$(1Ø)="
       "
4Ø9Ø FOR N=1 TO 1Ø: PRINT AT N+5
,6; INK 2;A$(N): NEXT N: FOR M=1
 TO 4: BEEP 1,-1Ø: BEEP 1,-5: NE
XT M
41ØØ PAPER 7: INK Ø: CLS
411Ø PRINT AT 4,Ø;"WILL YOU ADVA
NCE (Key ""1"") OR RETREAT? (Key
 ""Ø"")"
412Ø IF INKEY$="" THEN  GO TO 41
2Ø
4125 IF INKEY$="1" THEN  GO TO 4
2ØØ
413Ø IF INKEY$="Ø" THEN  GO TO 1
5ØØ
42ØØ PRINT AT 8,4;" ";AT 9,4;"
 ";AT 11,4;" ";AT 12,4;" "
421Ø PRINT AT 1Ø,Ø;"
                   "
422Ø PRINT AT 14,Ø;"YOU SEE A SW
ORD LEFT BY A        PREVIOUS ADV
ENTURER.": PAUSE 1ØØ
423Ø IF RND>.5 THEN  PRINT AT 17
,Ø;"UNFORTUNATELY THE DRAGON'S F
IRE BURNS YOU TO DEATH AS YOU
    PICK UP THE SWORD!": FOR N=3
Ø TO -15 STEP -1: BEEP .1,N: NEX
T N: BEEP 1,-15: GO TO 85ØØ
424Ø PRINT AT 17,Ø;"YOU STAB THE
 DRAGON IN IT'S      HEART.": BEE
P 3,-2Ø: PRINT AT 19,Ø;"BEHIND T
HE BODY IS A CORRIDOR   OPEN THE
 DOOR AT THE END...": PAUSE 2ØØ:
 GO TO 2ØØØ
7999 REM ###GRAPHICS###
8ØØØ FOR M=Ø TO 5: FOR N=Ø TO 7
8Ø1Ø READ C: POKE USR CHR$ (144+
M)+N,C
8Ø2Ø NEXT N: NEXT M
8Ø3Ø DATA 247,247,247,247,247,24
7,247,247,128,192,224,24Ø,224,19
2,128,Ø,1,3,7,15,15,7,3,1
8Ø4Ø DATA 24,24,6Ø,9Ø,24,36,36,6
6,6Ø,24,6Ø,126,255,255,126,6Ø,32
,23Ø,226,62,62,34,33,129
8Ø5Ø RETURN
8499 REM ###AGAIN?###
85ØØ PAPER 7: CLS : BORDER 3: IN
K Ø
851Ø PRINT AT 1Ø,1;"DO YOU WANT
TO PLAY AGAIN"
852Ø IF INKEY$="" THEN  GO TO 85
2Ø
853Ø IF INKEY$="Y" THEN  BORDER
6: CLS : GO TO 499
854Ø IF INKEY$="N" THEN  CLS : P
RINT AT 8,4;"THANK YOU FOR PLAYI
NG";AT 1Ø,1Ø;"GOODBYE"
855Ø STOP
8599 REM ###TITLE###
86ØØ INK Ø: PRINT AT 9,11; PAPER
 6; BRIGHT 1;"
 "
861Ø PRINT AT 1Ø,11; PAPER 6; BR
IGHT 1;"          "
862Ø PRINT AT 11,11; PAPER 6; BR
IGHT 1;"          "
863Ø RETURN
```

Poojan

Pim Quakkelaar gives us the Netherlands version of 'The Three Little Pigs.'

It is the nature of things that wolves try to eat the pigs and the pigs, not surprisingly, are none too keen on the idea.

The wolves though, are getting clever and here we find them trying two different ways of achieving their aim, first by ballooning out of a tree to the ground to reach the piglets, and then, if this fails, ballooning up into the tree in order to push boulders on top of you.

Of course, as parent pig you're not going to stand by and let all this happen and are positioned in your vertically moving cage armed with a bow and arrows with which to puncture the wolve's hopes, and balloons.

We'll let Pim explain the whole sordid thing in glorious technicolour detail . . . . .

Your Task

After the introduction, the control keys are displayed and you move into the first stage of the game. The program can continue indefinitely, but consists of three main screens. Using your bow and arrow, you must shoot the wolf's balloon to prevent him from reaching the piglets. If the wolf manages to land safely he will eat one of them.

On the second screen you have to shoot as many balloons as possible, and in the third you also have to face the rocks that the wolf will try to throw at you.

The Score

On the first screen you earn points for hitting the wolf or the balloon. The wolf is worth ten points, but the score for the balloons depends on how far away they are. The higher up in the sky the balloon, the more points you score for hitting it.

The further you progress in the game, the harder the screens become.

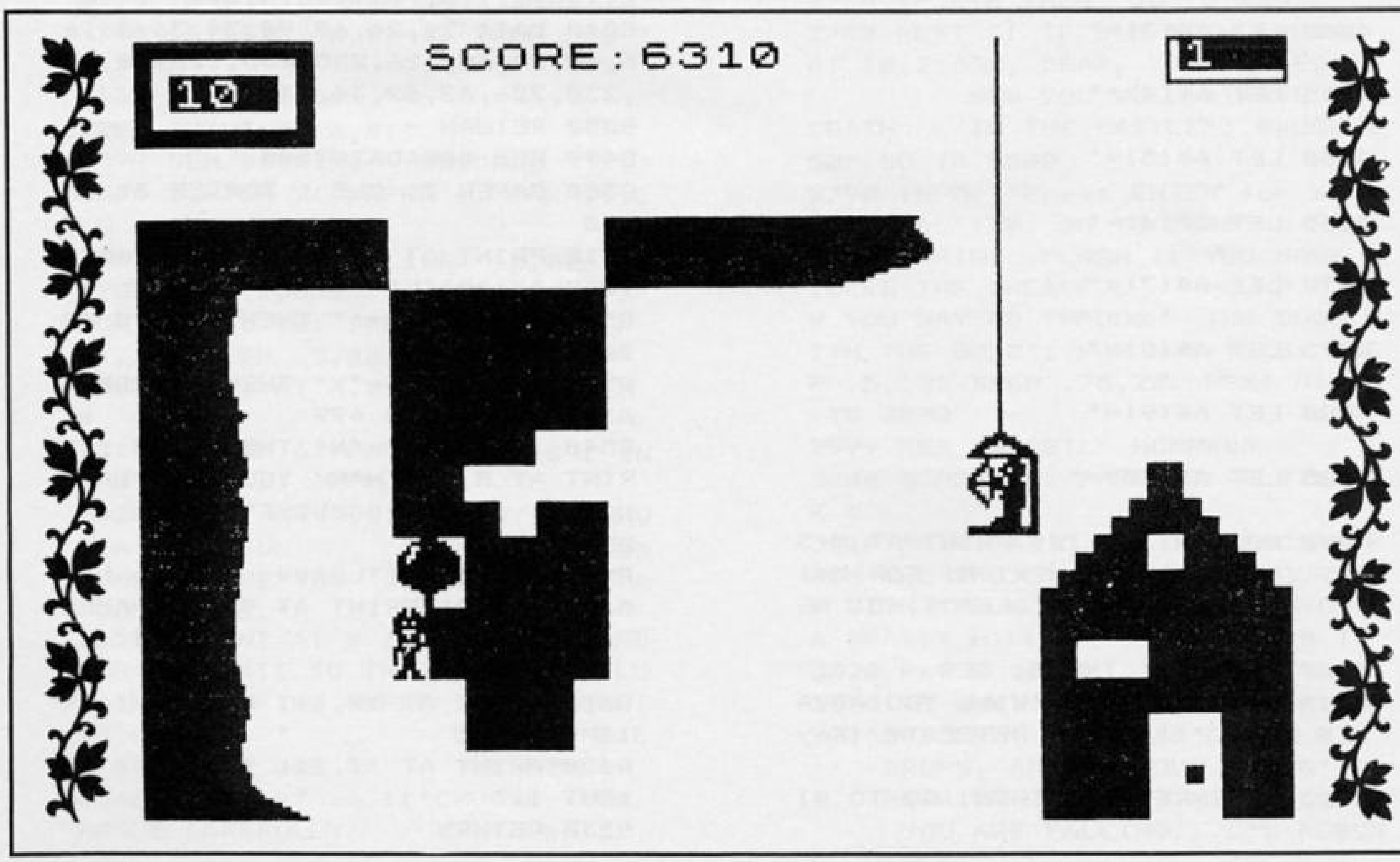
Program Details

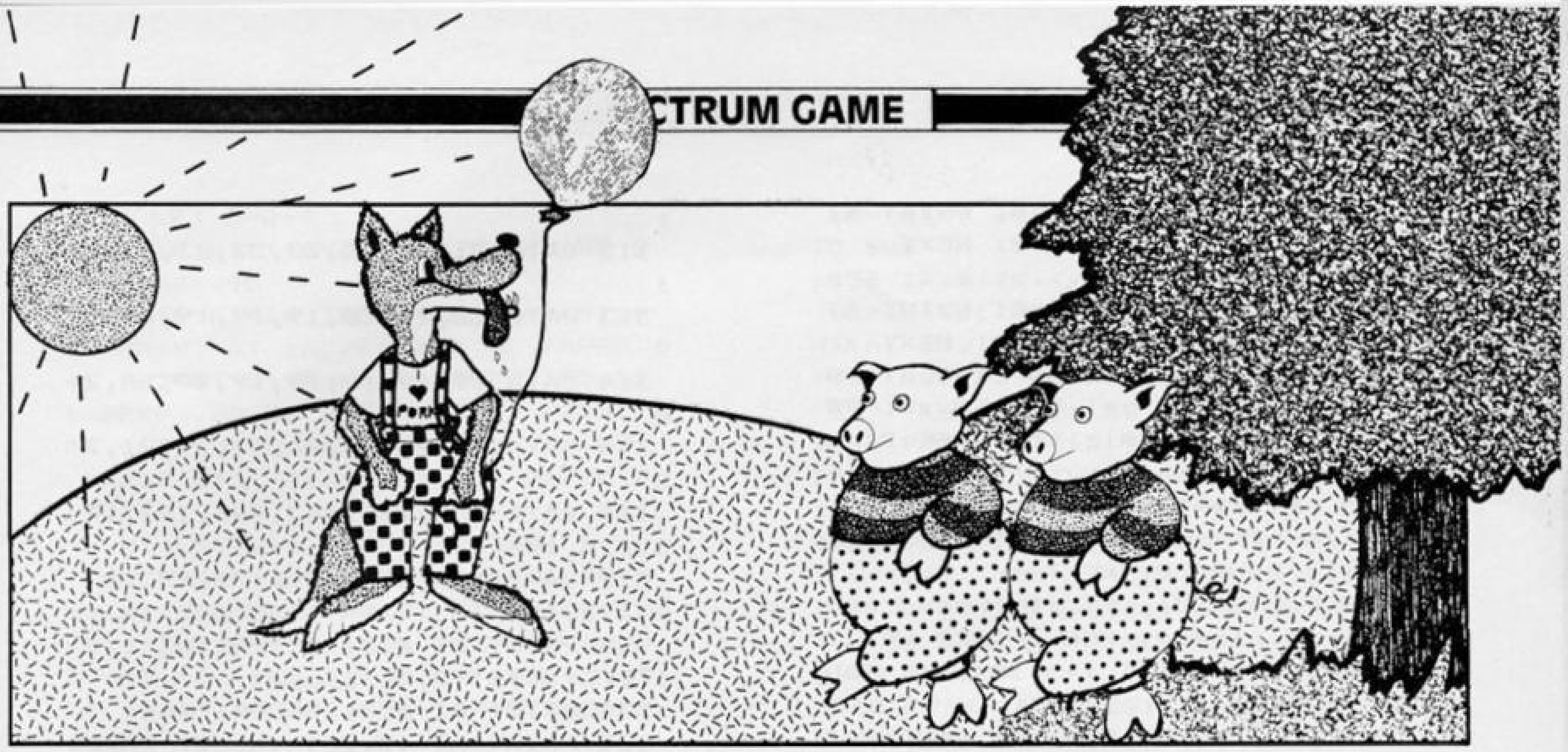
20-170	Introduction.
175-200	Control keys.
250	Variables.
255-2035	Screen1.
2400-2890	Screen2.
3000-8499	Screen3.
8500-8525	Tree for screens 2 & 3.
9000-9400	Hall of Fame (input routine).
9500-9565	UDGs.
9600-9840	Hall of Frame (calculations routine).

Variables

bon; the score at which you earn two extra pigs.
mnow; maximum number of wolves needed to throw the rock.
dis; the distance that the wolves jump from.
cag; number of cages that you own.
now;w;wl; the number of wolves on different screens.
pig; the number of pigs you have.
sc; score.
stage; stage counter.

Once you've got the program typed in, it can be saved with the command SAVE "Poojan" LINE 1.





```
   1 REM *********************
         *Underlined characters*
         *are entered in       *
         *GRAPHICS mode.       *
         *********************

  20 CLS : PRINT AT 10,0;"READIN
G GRAPHICS; HANG ON A SEC."
  30 GO SUB 9500
  40 INK 7: PAPER 0: BORDER 0: B
RIGHT 1: CLS
  45 LET z$="P.Q.PRODUCTIONS PRE
SENTS": FOR z=1 TO LEN z$: PRINT
 AT 3,z+3;z$(z): PAUSE 5: NEXT z
: RESTORE 45: FOR n=1 TO 3: READ
 t: BEEP .15,t: NEXT n: DATA 0,2
,4: BEEP .35,7: BEEP .20,4: BEEP
 .3,7
  50 RESTORE 50: DIM a(6): FOR t
=1 TO 6: READ a(t): NEXT t: DATA
 10,12,14,16,18,20: FOR x=1 TO 6
: FOR y=20 TO 10 STEP -1: PRINT
AT y,a(x); INK x+1;"CD";AT y-1,a
(x); INK x+1;"AB": BEEP .025,y/x
:: PRINT AT y,a(x);"  ": NEXT y:
 PRINT AT y+1,a(x); INK x+1;"CD"
: PRINT AT 3,4; INK X+1; BRIGHT
1;"P.Q.PRODUCTIONS PRESENTS": NE
XT x
  60 INK 6: BRIGHT 1: FOR x=1 TO
 6: PRINT AT 9,a(x);"RS";AT 10,a
(x);"TU": BEEP .01,0: BEEP .01,5
: BEEP .01,10: PRINT AT 9,a(x);"
  ";AT 10,a(x);"  ": GO SUB 100:
 NEXT x
  70 GO TO 165
 100 GO TO 100+(x*10)
 110 PLOT 84,90: DRAW 0,10: DRAW
 10,-1,-1.5: DRAW -7,-5,-1.5: DR
AW 0,2: DRAW 3,3,.5: RETURN
 120 CIRCLE 102,95,6: CIRCLE 102
,95,4: RETURN
 130 CIRCLE 118,95,6: CIRCLE 118
,95,4: RETURN
 140 PLOT 128,88: DRAW 5,0: DRAW
 5,13,1.5: DRAW -7,0: DRAW 0,-2:
 DRAW 5,0: DRAW -5,-9,-2.5: DRAW
 -3,0: DRAW 0,-2: RETURN
 150 PLOT 144,90: DRAW 0,8: DRAW
 14,0,-2: DRAW 0,-8: DRAW -3,0,-
2: DRAW 0,4: DRAW -8,0: DRAW 0,-
4: DRAW -2,0,-2: PLOT 147,97: DR
AW 6,0: DRAW -6,0,1: RETURN
 160 PLOT 163,92: DRAW 0,10: DRA
W 3,0,-1: DRAW 10,-10,2: DRAW 0,
10: DRAW 3,0,-1: DRAW 0,-13: DRA
W -3,0: DRAW -10,5,-2: DRAW 0,-5
: DRAW -3,0,-2: DRAW 0,3: RETURN

 165 FOR t=1 TO 2: FOR y=0 TO 1:
 FOR x=1 TO 7: PLOT  INK x; BRIG
HT y;56,72: DRAW  INK x; BRIGHT
y;150,0: BEEP .02,0: DRAW  INK x
; BRIGHT y;0,48: BEEP .02,7: DRA
W  INK x; BRIGHT y;-150,0: BEEP
.02,9: DRAW  INK x; BRIGHT y;0,-
48: BEEP .02,5: NEXT x: NEXT y:
NEXT t
 170 BEEP 1,0: BEEP 1,-12
 175 DIM s(10): DIM n$(10,20): F
OR t=1 TO 10: LET n$(t)=" ??????
???????????? ": NEXT t
 180 INK 6: BRIGHT 1: CLS
 185 LET z$="CONTROL": FOR z=1 T
O LEN z$: PRINT AT 2,z+12;z$(z);
: PAUSE 2: NEXT z
 187 LET z$="UP    .......... KEY
 A": FOR z=1 TO LEN z$: PRINT AT
 6,z+5;z$(z);: PAUSE 2: NEXT z
 189 LET z$="DOWN .......... KEY
```

```
 Z": FOR z=1 TO LEN z$: PRINT AT
 8,z+5;z$(z);: PAUSE 2: NEXT z
 191 LET z$="FIRE .......... SPA
CE": FOR z=1 TO LEN z$: PRINT AT
 1Ø,z+5;z$(z);: PAUSE 2: NEXT z
 193 LET z$=" 2 BONUS PIGS AT 1Ø
,ØØØ AND": FOR z=1 TO LEN z$: PR
INT AT 19,z+1;z$(z);: PAUSE 2: N
EXT z
 195 LET z$="EVERY 2Ø,ØØØ": FOR
z=1 TO LEN z$: PRINT AT 21,z+9;z
$(z);: PAUSE 2: NEXT z
 199 PAUSE 5Ø: LET z$="Press any
 key to play.": FOR z=1 TO LEN z
$: PRINT #1;AT 1,z+5;z$(z);: PAU
SE 2: NEXT z
 200 FOR k=Ø TO 8ØØ: IF INKEY$="
" THEN  NEXT k: GO TO 98ØØ
 25Ø LET bon=1ØØØØ: LET mnow=5:
LET dis=12: LET cag=2: LET now=1
9: LET pig=1Ø: LET sc=Ø: LET w=1
Ø: LET stage=1: LET w1=5
 255 INK 5: BORDER Ø: BRIGHT 1:
PAPER Ø: CLS
 28Ø LET w1=w1+5
 285 LET w=w1+5: IF w>4Ø THEN  L
ET w=4Ø
 29Ø LET dis=dis-2
 295 IF dis<=6 THEN  LET dis=6
 297 LET z$="STAGE ": FOR z=1 TO
 LEN z$: PRINT AT 1Ø,11+z;z$(z);
: PAUSE 3: NEXT z: PRINT stage
 3ØØ RESTORE 3Ø1: FOR x=Ø TO 135
: READ a: PLOT  INK 4; BRIGHT 1;
Ø,x: DRAW  INK 4; BRIGHT 1;a,Ø:
NEXT x
 3Ø1 DATA 36,34,32,3Ø,28,26,25,2
4
 3Ø2 DATA 23,23,23,22,22,21,21,2
1
 3Ø3 DATA 21,21,22,22,22,22,22,2
3
 3Ø4 DATA 23,23,23,23,22,22,22,2
1
 3Ø5 DATA 21,21,21,2Ø,2Ø,2Ø,2Ø,2
Ø
 3Ø6 DATA 2Ø,2Ø,2Ø,2Ø,2Ø,2Ø,2Ø,2
Ø
 3Ø7 DATA 2Ø,2Ø,21,21,21,21,21,2
1
 3Ø8 DATA 21,22,22,22,22,22,22,2
3
 3Ø9 DATA 23,23,23,23,23,23,23,2
2
 31Ø DATA 22,21,21,22,22,23,23,2
3
 311 DATA 23,23,22,22,22,22,21,2
1
 312 DATA 21,21,22,23,22,21,21,2
1
 313 DATA 2Ø,2Ø,2Ø,19,19,19,19,1
9
 314 DATA 19,19,19,19,19,19,2Ø,2
Ø
 315 DATA 2Ø,2Ø,21,21,21,21,21,2
2
 316 DATA 156,161,163,166,171,17
3,175,176
 317 DATA 176,176,175,174,173,17
2,172,173
 325 PRINT AT Ø,Ø;"▛▀▀▀▀▀▀▜";AT 1,Ø
;"▌      ▐";AT 2,Ø;"▙▄▄▄▄▄▄▟"
 33Ø PRINT AT Ø,8;"SCORE:";sc
 335 PRINT AT 12,25; INK 2; PAPE
R Ø;"    █";AT 13,25;"  ▄██▄";AT 1
4,25;" ▄█ █▄";AT 15,25;"▄█████▄"
;AT 16,25; PAPER 1; INK 4; BRIGH
T Ø;"███████";AT 17,25;"█  ████"
;AT 18,25;"███████";AT 19,25;"██
█  ██";AT 20,25;"███ .██";AT 21,
25;"███  ██"
 34Ø FOR x=Ø TO 7: PRINT AT x,23
; INK 6;"E": NEXT x
 345 PLOT 229,165: DRAW 26,Ø: DR
AW Ø,1Ø: DRAW -26,Ø: DRAW Ø,-1Ø:
 PRINT AT Ø,29; INVERSE 1; PAPER
 1;"  "
 35Ø LET stage=stage+1: LET u=Ø:
 LET a$="FG": LET x2=2: LET y=22
: LET x=7
 4ØØ LET z$="READY ?": FOR z=1 T
O LEN z$: PRINT AT 12,z+11;z$(z)
: PAUSE 3: NEXT z
 41Ø RESTORE 41Ø: FOR n=1 TO 12:
 READ t: BEEP .3Ø,t: NEXT n: DAT
A 12,1Ø,9,7,5,2,Ø,Ø,2,5,4,7: BEE
P .7Ø,5: BEEP .3Ø,5
 42Ø PRINT AT 1Ø,12;"        ";A
T 12,12;"       "
 5ØØ LET i=INT (RND*5+1.5)
 5Ø1 IF i=5 THEN  GO TO 5ØØ
 5Ø2 LET x3=INT ((dis+RND*4)+.5)
 5Ø3 PRINT AT 1,1; INVERSE 1; PA
PER 1;w;"  "
 5Ø5 IF sc>=bon THEN  LET pig=pi
g+2: LET bon=bon+2ØØØØ
 51Ø PRINT AT Ø,14;sc;AT Ø,29; I
NVERSE 1;pig;"  "
 515 IF pig=Ø THEN  GO TO 96ØØ
 52Ø IF w=Ø THEN  PAUSE 5Ø: GO T
O 24ØØ
 55Ø FOR t=1 TO x3
 552 LET t2=t-1
 555 IF t<>x3 THEN  GO TO 75Ø
```

```
 575 LET u=u+1
 578 IF u>=1 AND u<=7 THEN  GO S
UB 2ØØØ+u*5
 58Ø IF u>=19 THEN  LET pig=pig-
1: PRINT AT 16,26; INK Ø; PAPER
4; BRIGHT Ø;"MAMA!": BEEP .Ø5,12
: BEEP .Ø5,6: BEEP .Ø5,Ø: PRINT
AT u-1,t2; PAPER Ø;"  ";AT u,t2;
 PAPER Ø;"  ";AT u+1,t2; PAPER Ø
;"  ";AT u+2,t2; PAPER Ø;"  ": LET
 u=Ø: LET w=w-1: GO TO 5ØØ
 6ØØ PRINT AT 16,25; INK 4; BRIG
HT Ø;"■■■■■■": PRINT AT u,t2; IN
K i;"AB";AT u+1,t2; INK i;"CD";A
T u+2,t2; INK 7;"J";AT u+3,t2; I
NK 7;"K";AT u-1,t2; INK Ø;"■■":
GO TO 76Ø
 75Ø PRINT AT 3,t; INK 7;"H";AT
4,t; INK 7;"I";AT 3,t-1;" ";AT 4
,t-1;" "
 755 IF INKEY$="" THEN  GO TO 85
Ø
 765 IF INKEY$="a" THEN  PRINT A
T x+2,23;"  ";AT x+1,21;"  ": LE
T x=x-1
 77Ø IF x<=7 THEN  LET x=7
 78Ø IF INKEY$="z" THEN  PRINT A
T x,23;"E ";AT x+1,21;"  ": LET
x=x+1
 785 IF x>=19 THEN  LET x=19
 81Ø IF y<>22 THEN  GO TO 85Ø
 82Ø IF INKEY$=" " THEN  LET x2=
x+1: LET y=y-1: LET a$=""
 85Ø PRINT AT x,23; INK 6;"LM";A
T x+1,23; INK 6;"NO";AT x+2,23;
INK 6;"PQ";AT x-1,23; INK 6;"E";
AT x+1,21; INK 5;a$
 86Ø IF t=x3 AND y<>22 THEN  GO
TO 89Ø
 87Ø IF t=x3 THEN  GO TO 575
 88Ø IF y=22 THEN  NEXT t
 9ØØ LET y=y-2
1ØØØ IF y<=2 THEN  PRINT AT x2,3
;"  ": LET y=22: LET x2=2: LET a
$="FG": GO TO 1Ø5Ø
1Ø1Ø PRINT AT x2,y+2;"  "
1Ø2Ø IF ATTR (x2,y)=64+i THEN  P
RINT AT u,t2;"RS";AT u+1,t2;"TU"
: BEEP .Ø1,Ø: BEEP .Ø1,5: BEEP .
Ø1,1Ø: LET w=w-1: LET sc=sc+(2Ø-
u)*1Ø: PRINT AT Ø,14;sc: PRINT A
T u,t2;"    ";AT u+1,t2;"    ": FO
R z=u+2 TO 2Ø: PRINT AT z,t2; IN
K 2; BRIGHT 1;"J";AT z+1,t2; INK
 2; BRIGHT 1;"K";AT z-1,t2; INK
Ø;"■": NEXT z: PRINT AT z-1,t2;"
 ";AT z,t2;" ": BEEP .Ø1,-1: LET
 a$="FG": LET y=22: LET u=Ø: GO
TO 5ØØ
1Ø3Ø IF ATTR (x2,y)=71 THEN  LET
 sc=sc+1Ø: PRINT AT Ø,14;sc: PRI
NT AT u+2,y;"  ";AT u+3,y;"  ":
BEEP .Ø1,6: BEEP .Ø1,12: LET a$=
"FG": LET y=22: GO TO 575
1Ø45 PRINT AT x2,y; INK 5;"FG"
1Ø5Ø IF t=x3 THEN  GO TO 575
1Ø6Ø NEXT t
2ØØ5 PRINT AT u,t2; INK i;"AB";A
T u+1,t2; INK i;"CD";AT u+2,t2;
INK 7;"J";AT u+3,t2; INK 7;"K";A
T Ø,5; INK 5;"   SCORE:";sc;"  "
: GO TO 76Ø
2Ø1Ø PRINT AT u,t2; INK i;"AB";A
T u+1,t2; INK i;"CD";AT u+2,t2;
INK 7;"J";AT u+3,t2; INK 7; PAPE
R 4;"K";AT u-1,t2; PAPER Ø;"  ":
 GO TO 76Ø
2Ø15 PRINT AT u,t2; INK i;"AB";A
T u+1,t2; INK i;"CD";AT u+2,t2;
INK 7; PAPER 4;"J";AT u+3,t2; IN
K 7; PAPER 4;"K";AT u-1,t2; PAPE
R Ø;"  ": GO TO 76Ø
2Ø2Ø PRINT AT u,t2; INK i;"AB";A
T u+1,t2; INK i; PAPER 4;"CD";AT
 u+2,t2; INK 7; PAPER 4;"J";AT u
+3,t2; INK 7; PAPER Ø;"K";AT u-1
,t2; PAPER Ø;"  ": GO TO 76Ø
2Ø25 PRINT AT u,t2; INK i; PAPER
 4;"AB";AT u+1,t2; INK i; PAPER
4;"CD";AT u+2,t2; INK 7; PAPER Ø
;"J";AT u+3,t2; INK 7; PAPER Ø;"
K";AT u-1,t2; PAPER Ø;"  ": GO T
O 76Ø
2Ø3Ø PRINT AT u,t2; INK i; PAPER
 4;"AB";AT u+1,t2; INK i; PAPER
Ø;"CD";AT u+2,t2; INK 7; PAPER Ø
;"J";AT u+3,t2; INK 7; PAPER Ø;"
K";AT u-1,t2; PAPER 4;"  ": GO T
O 76Ø
2Ø35 PRINT AT u,t2; INK i; PAPER
 Ø;"AB";AT u+1,t2; INK i; PAPER
Ø;"CD";AT u+2,t2; INK 7; PAPER Ø
;"J";AT u+3,t2; INK 7; PAPER Ø;"
K";AT u-1,t2; PAPER 4;"  ": GO T
O 76Ø
24ØØ INK 5: PAPER Ø: BORDER Ø: B
RIGHT 1: CLS : LET z$="CHANCE ST
AGE": FOR z=1 TO LEN z$: PRINT A
T 1Ø,z+8;z$(z);: PAUSE 3: NEXT z
: GO SUB 85ØØ
24Ø3 DIM b(1Ø,2)
24Ø5 FOR t=1 TO 1Ø
241Ø FOR r=1 TO 2
2415 LET b(t,r)=INT ((dis+RND*4)
```

```
+.5)
2420 NEXT r
2425 IF b(t,1)=b(t,2) THEN  GO T
O 2410
2430 NEXT t
2506 PRINT AT 0,0;"▛▀▀▀▀▜";AT 1,0
;"▌    ▐";AT 2,0;"▙▄▄▄▄▟"
2507 FOR x=0 TO 7: PRINT AT x,23
; INK 6;"E": NEXT x
2510 PRINT AT 12,25; INK 2; PAPE
R 0;"   ▀";AT 13,25;"  ▟█▙";AT 1
4,25;" ▟█ ██▙";AT 15,25;"▟█████▙"
;AT 16,25; PAPER 1; INK 4; BRIGH
T 0;"███████";AT 17,25;"█  ███"
;AT 18,25;"███████";AT 19,25;"██
█ ██";AT 20,25;"███ ▄██";AT 21,
25;"███  ██"
2515 PRINT AT 0,8;"SCORE:";sc
2517 LET z$="READY ?": FOR z=1 T
O LEN z$: PRINT AT 12,11+z;z$(z)
;: PAUSE 3: NEXT z: LET stage=st
age+1
2518 RESTORE 2518: FOR n=1 TO 13
: READ t: BEEP .30,t: NEXT n: DA
TA 0,4,0,5,0,7,5,4,4,5,2,0,-1: B
EEP .45,0: PRINT AT 10,8;"
  ";AT 12,12;"       "
2520 LET hit=0: LET r=0: LET cou
nt=10: LET x2=2: LET y=22: LET x
=5: LET a$="FG"
2525 LET r=r+1: LET vv=1: LET v=
1: LET y21=20.6: LET y22=20.3
2526 LET i1=INT (RND*6+1.5): IF
i1=5 THEN  GO TO 2526
2527 LET i2=INT (RND*6+1.5): IF
i2=5 THEN  GO TO 2526
2528 IF i1=i2 THEN  GO TO 2527
2530 IF r=11 THEN  GO TO 2850
2532 IF sc>=bon THEN  LET pig=pi
g+2: LET bon=bon+20000
2535 PRINT AT 1,1; INVERSE 1; IN
K 5;11-r;" "
2555 IF v=0 AND vv=0 THEN  GO TO
 2525
2557 IF v=0 THEN  GO TO 2575
2560 LET y21=y21-.8
2565 IF INT (y21+.5)<=5 THEN  PR
INT AT 5,b(r,1);"RS";AT 6,b(r,1)
;"TU": BEEP .01,0: BEEP .01,5: B
EEP .01,10: PRINT AT 5,b(r,1);"
 ";AT 6,b(r,1);"  ": LET v=0: GO
 TO 2572
2570 PRINT AT y21,b(r,1); INK i1
;"CD";AT y21-1,b(r,1); INK i1;"A
B";AT y21+1,b(r,1); INK 0;"██"
2572 IF vv=0 THEN  GO TO 2600
2575 LET y22=y22-.5
2580 PRINT AT y22,b(r,2); INK i2
;"CD";AT y22-1,b(r,2); INK i2;"A
B";AT y22+1,b(r,2); INK 0;"██"
2585 IF INT (y22+.5)<=6 THEN  PR
INT AT 5,b(r,2);"RS";AT 6,b(r,2)
;"TU": BEEP .01,0: BEEP .01,5: B
EEP .01,10: PRINT AT 5,b(r,2);"
 ";AT 6,b(r,2);"  ": LET y21=20:
 LET y22=20: GO TO 2525
2590 IF INKEY$="" THEN  GO TO 27
00
2600 IF INKEY$="a" THEN  PRINT A
T x+2,23;"  ";AT x+1,21;"  ": LE
T x=x-1
2605 IF x<=5 THEN  LET x=5
2610 IF INKEY$="z" THEN  PRINT A
T x,23;"  ";AT x+1,21;"  ": LET
x=x+1
2615 IF x>=19 THEN  LET x=19
2617 IF y<>22 THEN  GO TO 2700
2620 IF INKEY$=" " THEN  LET x2=
x+1: LET a$="": LET y=y-1
2700 PRINT AT x,23; INK 6;"LM";A
T x+1,23; INK 6;"NO";AT x+2,23;
INK 6;"PQ";AT x-1,23; INK 6;"E";
AT x+1,21; INK 5;a$
2800 IF y=22 THEN  GO TO 2550
2802 LET y=y-2
2803 IF y<=2 THEN  PRINT AT x2,3
;"  ": LET a$="FG": LET y=22: GO
 TO 2550
2804 PRINT AT x2,y+2;"  "
2807 IF ATTR (x2,y)=64+i2 THEN
PRINT AT y22-1,b(r,2);"RS";AT y2
2,b(r,2);"TU";AT x2,y+2;"  ": BE
EP .01,0: BEEP .01,5: BEEP .01,1
0: PRINT AT y22-1,b(r,2);"  ";AT
 y22,b(r,2);"  ": LET y=22: LET
a$="FG": LET vv=0: LET sc=sc+b(r
,2)*15: PRINT AT 0,14;sc: LET hi
t=hit+1: GO TO 2555
2810 IF ATTR (x2,y)=64+i1 THEN
PRINT AT y21-1,b(r,1);"RS";AT y2
1,b(r,1);"TU";AT x2,y+2;"  ": BE
EP .01,0: BEEP .01,5: BEEP .01,1
0: PRINT AT y21-1,b(r,1);"  ";AT
 y21,b(r,1);"  ": LET y=22: LET
a$="FG": LET v=0: LET sc=sc+b(r,
2)*25: PRINT AT 0,14;sc: LET hit
=hit+1: GO TO 2555
2815 PRINT AT x2,y; INK 5;"FG"
2825 GO TO 2550
2850 FOR g=0 TO 100: CLS
2855 LET z$="NUMBER OF HITS :":
FOR g=1 TO LEN z$: PRINT AT 8,g+
6;z$(g);: PAUSE 3: NEXT g: PRINT
  INK 7;hit
```

```
2860 LET z$="BONUS : 100x": FOR
g=1 TO LEN z$: PRINT AT 10,g+5;z
$(g);: PAUSE 3: NEXT g: PRINT ;h
it;"= "; INK 7;100*hit
2870 IF hit=20 THEN  LET z$="EXT
RA BONUS : 25,000": FOR g=1 TO L
EN z$: PRINT AT 13,g+5; INK 6;z$
(g): BEEP .07,CODE z$(g)-60: NEX
T g: LET sc=sc+25000
2880 LET sc=sc+hit*100
2890 RESTORE 2890: FOR n=1 TO 8:
 READ t: BEEP .35,t: NEXT n: DAT
A 12,9,7,9,12,9,7,9: BEEP .5,5
3000 INK 5: CLS : LET z$="STAGE
": FOR z=1 TO LEN z$: PRINT AT 1
0,11+z;z$(z);: PAUSE 3: NEXT z:
PRINT stage: LET stage=stage+1:
GO SUB 8500
3010 LET now=19: LET w=w1+5: LET
 x2=2: LET x=5
3100 PRINT AT 0,0;"▛▀▀▀▀▜";AT 1,0
;"▌    ▐";AT 2,0;"▙▄▄▄▄▟"
3105 FOR x=0 TO 7: PRINT AT x,23
; INK 6;"E": NEXT x
3110 PRINT AT 12,25; INK 2; PAPE
R 0;"   ■";AT 13,25;"  ▄█▙";AT 1
4,25;" ▄█ █▙";AT 15,25;"▄████▙"
;AT 16,25; PAPER 1; INK 4; BRIGH
T 0;"██████";AT 17,25;"█ ███"
;AT 18,25;"██████";AT 19,25;"██
█ ██";AT 20,25;"███ ▄ ██";AT 21,
25;"███ ██"
3115 PRINT AT 0,8;"SCORE:";sc
3120 PRINT AT 1,19; INK 6;"▗█▙ ";
AT 2,19; INK 6;"▝▀▘"
3130 PLOT 229,165: DRAW 26,0: DR
AW 0,10: DRAW -26,0: DRAW 0,-10:
 PRINT AT 0,29; INVERSE 1; PAPER
 1;"  "
3140 LET z$="READY ?": FOR z=1 T
O LEN z$: PRINT AT 12,11+z;z$(z)
;: PAUSE 3: NEXT z: FOR t=0 TO 8
0: NEXT t: PRINT AT 12,12;"
  ";AT 10,12;"       "
3150 LET y2=21: LET a$="FG": LET
 y=22
3152 LET x3=INT ((dis+RND*4)+.5)
3155 LET i=INT (RND*5+1.5): IF i
=5 THEN  GO TO 3155
3157 IF sc>=bon THEN  LET pig=pi
g+2: LET bon=bon+20000
3160 PRINT AT 1,1; INVERSE 1; PA
PER 1;w;" ";AT 0,29;mnow-(19-now
)
3165 IF 19-now=mnow THEN  GO TO
5500
3170 IF w=0 THEN  GO TO 5557
3180 PRINT AT y2-1,x3; INK 7;"J"
;AT y2,x3; INK 7;"K": GO TO 3500
3200 LET y2=y2-1
3210 IF y2<=7 THEN  GO TO 5000+y
2*5
3220 PRINT AT y2-3,x3; INK i;"AB
";AT y2-2,x3; INK i;"CD";AT y2-1
,x3; INK 7;"J";AT y2,x3; INK 7;"
K";AT y2+1,x3; INK 0;"■";AT y2-1
,x3+1;"■"
3400 IF INKEY$="" THEN  GO TO 36
00
3500 IF INKEY$="a" THEN  PRINT A
T x+2,23;"  ";AT x+1,21;"  ": LE
T x=x-1
3505 IF x<=5 THEN  LET x=5
3510 IF INKEY$="z" THEN  PRINT A
T x,23;"  ";AT x+1,21;"  ": LET
x=x+1
3515 IF x>=19 THEN  LET x=19
3520 IF y<>22 THEN  GO TO 3600
3530 IF INKEY$=" " THEN  LET x2=
x+1: LET a$="": LET y=y-1
3600 PRINT AT x,23; INK 6;"LM";A
T x+1,23; INK 6;"NO";AT x+2,23;
INK 6;"PQ";AT x-1,23; INK 6;"E";
AT x+1,21; INK 5;a$
3610 IF y=22 THEN  GO TO 3200
3650 LET y=y-2
3655 IF y<=2 THEN  PRINT AT x2,3
;"  ": LET a$="FG": LET y=22: GO
 TO 3200
3660 PRINT AT x2,y+2;"  "
3670 IF ATTR (x2,y)=64+i THEN  P
RINT AT y2-3,x3;"RS";AT y2-2,x3;
"TU": BEEP .01,0: BEEP .01,5: BE
EP .01,10: PRINT AT y2-3,x3; INK
 0;"■■";AT y2-2,x3;"■■": FOR t=y
2 TO 20: PRINT AT t,x3; INK 2;"J
";AT t+1,x3; INK 2;"K";AT t-1,x3
; INK 0;"■": NEXT t: BEEP .01,-1
: PRINT AT 20,x3; INK 0;"■";AT 2
1,x3;"■": LET w=w-1: LET sc=sc+y
2*10: PRINT AT 0,14;sc: GO TO 31
50
3680 IF ATTR (x2,y)=71 THEN  BEE
P .01,6: BEEP .01,12: PRINT AT y
2,x3; INK 0;"■■": LET sc=sc+10:
LET a$="FG": LET y=22: GO TO 320
0
3900 PRINT AT x2,y; INK 5;"FG"
4000 GO TO 3200
5010 PRINT AT 1,x3; INK 7;"J";AT
 2,x3;"K";AT 3,x3; INK 4;"■";AT
1,x3+1; INK 0;"■";AT 1,x3; INK 0
;"■";AT 2,x3; INK 0;"■": LET now
=now-1: LET w=w-1: FOR t=now TO
```

▶

```
18: PRINT AT 1,t;"H";AT 2,t;"I":
 NEXT t: LET y2=21: GO TO 3152
5015 PRINT AT 1,x3; INK i;"CD";A
T 2,x3; INK 7;"J";AT 3,x3; PAPER
 4;"K";AT 4,x3; INK 4;"■";AT 2,x
3+1; INK Ø;"■": GO TO 35ØØ
5Ø2Ø PRINT AT 1,x3; INK i;"AB";A
T 2,x3;"CD";AT 3,x3; PAPER 4; IN
K 7;"J";AT 4,x3;"K";AT 5,x3; INK
 Ø;"■";AT 3,x3+1; INK 4;"■": GO
TO 35ØØ
5Ø25 PRINT AT 2,x3; INK i;"AB";A
T 3,x3; PAPER 4;"CD";AT 4,x3; IN
K 7;"J";AT 5,x3; PAPER Ø;"K";AT
6,x3; INK Ø;"■";AT 4,x3+1; INK 4
;"■": GO TO 35ØØ
5Ø3Ø PRINT AT 3,x3; PAPER 4; INK
 i;"AB";AT 4,x3;"CD";AT 5,x3; PA
PER Ø; INK 7;"J";AT 6,x3;"K";AT
7,x3; INK Ø;"■";AT 5,x3+1;"■": G
O TO 35ØØ
5Ø35 PRINT AT 4,x3; PAPER 4; INK
 i;"AB";AT 5,x3; PAPER Ø;"CD";AT
 6,x3; INK 7;"J";AT 7,x3;"K";AT
8,x3; INK Ø;"■";AT 6,x3+1;"■": G
O TO 35ØØ
55ØØ PRINT AT 1,19;"   ";AT 2,19
;"   "
551Ø FOR t=1 TO x-2: PRINT AT t,
22; INK 6;"▄█▄ ";AT t+1,22; INK 6
;"▀█▀ ";AT t-1,22;" E ": BEEP .Ø1
,Ø: BEEP .Ø1,t: NEXT t
5512 PRINT AT t-1,22;"R:S";AT t,
22;"T:T": PAUSE 5
5515 PRINT AT t-1,22;"   ";AT t,
22;"   "
5517 PRINT AT x+1,21;"   "
552Ø FOR t=x+2 TO 21
554Ø PRINT AT t-2,23; INK 6;"LM"
;AT t-1,23; INK 6;"NO";AT t,23;
INK 6;"PQ";AT t-3,23;"  ": BEEP
.Ø1,6: BEEP .Ø1,2*t
5545 NEXT t
555Ø PRINT AT 19,23;"RS";AT 2Ø,2
3;"__";AT 21,23;"TU": BEEP .Ø1,Ø
: BEEP .Ø1,5: BEEP .Ø1,Ø
5555 PRINT AT 19,23;"  ";AT 2Ø,2
3;"  ";AT 21,23;"  ": PAUSE 5: L
ET cag=cag-1
5557 PAUSE 5Ø
556Ø CLS : PRINT AT Ø,1Ø;"SCORE:
";sc: LET z$="PIGS LEFT : ": FOR
 t=1 TO LEN z$: PRINT AT 1Ø,t+9;
z$(t);: PAUSE 5: NEXT t: PRINT p
ig
5565 LET z$="CAGES LEFT : ": FOR
 z=1 TO LEN z$: PRINT AT 12,z+8;
z$(z);: PAUSE 5: NEXT z: PRINT c
ag
5567 IF cag<=Ø THEN  PAUSE 2ØØ:
GO TO 96ØØ
557Ø PAUSE 1ØØ
5575 LET mnow=mnow-1: IF mnow<=3
 THEN  LET mnow=3
8499 GO TO 255
85ØØ RESTORE 85Ø5: FOR x=Ø TO 15
1: READ a: PLOT  INK 4; BRIGHT 1
;Ø,x: DRAW  INK 4; BRIGHT 1;a,Ø:
 NEXT x
85Ø5 DATA 36,34,32,3Ø,28,26,25,2
3,23,23,23,22,22,21,21,21,21,21,
22,22,22,22,22,23,23,23,23,23,22
,22,22,21,21,21,21,2Ø,2Ø,2Ø,2Ø,2
Ø,21,21,21,21,21,21,22,22,22,23,
23,23,22,23,22,21
851Ø DATA 2Ø,2Ø,2Ø,2Ø,2Ø,2Ø,2Ø,2
Ø,2Ø,2Ø,21,21,21,21,21,21,21,22,
22,22,22,22,22,23,23,23,23,23,23
,23,22,22,23,23,23,23,22,23,23,2
3,23,23,22,22,22,22,21,21,21,21,
22,23,22,21,21,21,2Ø,2Ø,2Ø,19,19
,19,19,19
8515 DATA 19,19,19,19,19,19,2Ø,2
Ø,2Ø,2Ø,21,21,21,21,21,21,156,16
1,163,166,171,173,175,176,176,17
6,175,174,173,172,172,173
8525 RETURN
9ØØØ DIM m$(6,3Ø): LET m$(2)="
  A B C D E F G H I J K": LET M$
(4)="     L M N O P Q R S T U V":
 LET M$(6)="    W X Y Z . ?   EN
D RUB": LET n=1: INK 6: BRIGHT 1
: CLS
9Ø1Ø FOR a=2 TO 6 STEP 2: FOR b=
1 TO 25: PRINT AT a,b;m$(a,b): N
EXT b: NEXT a
9Ø12 LET z$="YOU MAY ENTER YOUR
NAME.": FOR z=1 TO LEN z$: PRINT
 AT 18,z+3;z$(z);: PAUSE 2: NEXT
 z: LET z$="( USE UP/DOWN & FIRE
 )": FOR z=1 TO LEN z$: PRINT AT
 2Ø,z+4;z$(z);: PAUSE 2: NEXT z
9Ø15 PLOT 48,78: DRAW  INK 7;16Ø
,Ø:
9Ø2Ø LET x=2: LET y=5
9Ø3Ø PRINT AT x,y; OVER 1;"_"
9Ø37 PAUSE Ø
9Ø4Ø IF INKEY$<>"a" AND INKEY$<>
"z" AND INKEY$<>" " THEN  GO TO
9Ø37
9Ø42 IF INKEY$=" " THEN  BEEP .Ø
5,25: GO TO 93ØØ
9Ø5Ø IF INKEY$="a" THEN  LET y=y
+2
```

```
9Ø52 IF INKEY$="z" THEN  LET y=y
-2: GO TO 9Ø75
9Ø55 IF y>25 AND x=2 THEN  PRINT
 AT 2,25;"K": LET x=4: LET y=5:
GO TO 9Ø3Ø
9Ø6Ø IF y>25 AND x=4 THEN  PRINT
 AT 4,25;"V": LET x=6: LET y=5:
GO TO 9Ø3Ø
9Ø62 IF x=6 AND y=19 THEN  PRINT
 AT 6,17;" ": LET x=6: LET y=2Ø:
 GO TO 9Ø3Ø
9Ø64 IF x=6 AND y=22 THEN  PRINT
 AT 6,2Ø;"N": LET x=6: LET y=24:
 GO TO 9Ø3Ø
9Ø65 IF y=26 AND x=6 THEN  PRINT
 AT 6,24;"U": LET x=2: LET y=5:
GO TO 9Ø3Ø
9Ø7Ø PRINT AT x,y-2; OVER Ø;m$(x
,y-2): GO TO 9Ø3Ø
9Ø75 IF y<5 AND x=2 THEN  PRINT
AT 2,5;"A": LET y=24: LET x=6: G
O TO 9Ø3Ø
9Ø8Ø IF x=6 AND y=22 THEN  PRINT
 AT 6,24;"U": LET x=6: LET y=2Ø
9Ø85 IF x=6 AND y=18 THEN  PRINT
 AT 6,2Ø;"N": LET x=6: LET y=17
9Ø9Ø IF x=6 AND y<5 THEN  PRINT
AT 6,5;"W": LET x=4: LET y=25
9Ø95 IF x=4 AND y<5 THEN  PRINT
AT 4,5;"L": LET x=2: LET y=25
9285 PRINT AT x,y+2;m$(x,y+2)
929Ø GO TO 9Ø3Ø
93Ø5 IF x=6 AND y=2Ø THEN  GO TO
 98ØØ
931Ø IF x<>6 OR y<>24 THEN  GO T
O 9385
9311 LET n$(j,n)=" ": PRINT AT 1
1,5+n;n$(j,n)
9312 LET n=n-1: IF n<1 THEN  BEE
P .5,-5: BEEP .5,-12: LET n=1
9315 GO TO 9Ø37
9385 LET n=n+1
939Ø IF n>19 THEN  GO TO 98ØØ
9395 LET n$(j,n)=M$(x,y)
9397 PRINT AT 11,5+n;n$(j,n): GO
 TO 9Ø37
94ØØ GO TO 9Ø3Ø
9499 STOP
95ØØ RESTORE 95Ø5: FOR x=Ø TO 16
7: READ v: POKE USR "A"+x,v: NEX
T x
95Ø5 DATA 3,15,19,55,1Ø3,111,127
,127: REM 95Ø5 TO 952Ø > balloon
(A,B,C,D)
951Ø DATA 192,24Ø,248,252,254,25
4,254,254
9515 DATA 127,127,63,15,3,1,1,1
952Ø DATA 254,254,252,24Ø,192,17
8,128,128
9525 DATA 1,1,1,1,1,1,1,1: REM >
 rope (E)
953Ø DATA Ø,8,24,48,127,48,24,8:
 REM 953Ø TO 9535 > arrow (F,G)
9535 DATA Ø,1,2,4,255,4,2,1
954Ø DATA 16Ø,224,176,255,245,25
Ø,127,112,112,252,2Ø8,2Ø8,2Ø8,2Ø
Ø,254,254: REM > walking wolfe (
H,I)
9545 DATA 41,1Ø9,255,215,255,199
,1Ø9,125,71,252,184,184,184,56,1
Ø8,238: REM > falling wolve (J,K
)
955Ø DATA 1,3,15,3Ø,56,112,255,8
,Ø,128,224,24Ø,6Ø,3Ø,255,115,24,
53,1ØØ,199,255,199,1ØØ,53,251,18
7,251,115,123,251,251,251,24,8,Ø
,Ø,3,7,255,255,251,251,243,243,2
43,243,255,255
956Ø DATA Ø,8,24,24,56,1Ø4,Ø,Ø,2
4,28,3Ø,59,58,48,Ø,Ø,248,24Ø,97,
67,7,15,14,14,Ø,Ø,24,15,7,3,1,1
9565 RETURN
96ØØ CLS : PRINT AT 11,11;"GAME
OVER": PAUSE 15Ø
96Ø2 IF sc<s(1Ø) THEN  GO TO 98Ø
Ø
96Ø5 FOR j=1 TO 1Ø
961Ø IF sc<s(j) THEN  NEXT j
9615 FOR c=9 TO j STEP -1
962Ø LET n$(c+1)=n$(c): LET s(c+
1)=s(c): NEXT c
9625 LET s(j)=sc
9627 LET n$(j)=""
963Ø GO TO 9ØØØ
98ØØ CLS
98Ø5 PRINT AT 3,1Ø;"HALL OF FAME
"
981Ø FOR z=1 TO 1Ø: PRINT AT z+7
,Ø;n$(z);" -  ";s(z): NEXT z
9815 PLOT Ø,Ø: DRAW  INK 5;255,Ø
: DRAW  INK 6;Ø,175: DRAW  INK 7
;-255,Ø: DRAW  INK 6;Ø,-175
982Ø PAUSE 5Ø: LET z$="Press any
 key to play.": FOR z=1 TO LEN z
$: PRINT #1;AT 1,z+5;z$(z);: NEX
T z
983Ø FOR k=1 TO 5: FOR c=1 TO 7:
 PLOT  INK c;Ø,Ø: DRAW  INK c;25
5,Ø: DRAW  INK c;Ø,175: DRAW  IN
K c;-255,Ø: DRAW  INK c;Ø,-175
9835 IF INKEY$<>"" THEN  GO TO 9
84Ø
9837 NEXT c: NEXT k: GO TO 18Ø
984Ø GO TO 23Ø
```

Scrabble Board

Robin Thompson takes the tedium out of the game with this ZX81 version for 2-4 players.

On RUNing, the program will ask how many people are to play, and will request their names. There will then be a brief pause while the board is set up and displayed. Instructions will be printed, and the game will commence, ending when all the letters have been used and no-one can continue.

Please note that the program has no facilities for correcting a mistake in the players' input, so care must be taken when typing in words.

Typing the program in should present no problems, other than the graphics characters, and these are listed in figure 1.

Variables

The main variables used by the program are as follows;
B$ -holds the board.
C$ -holds the letters to be distributed amongst the payers.
E$ -holds the players' letters.
X$ -holds the players' names.
X-holds the players' scores.
S1 -the start of the diaplay file.
P-the number of players.

Figure 1. Scrabble board graphics.

Line 11	*eight graphic 'S'*	*Line 8430*	*inverse 'T'*
Line 75	*inverse 'space'*	*Line 8435*	*inverse 'T'*
Line 7000	*inverse 'T'*	*Line 8445*	*inverse 'T'*
Line 7010	*inverse 'D'*	*Line 8450*	*inverse 'T'*
Line 8395	*inverse 'space'*	*Line 8460*	*inverse 'D'*
Line 8400	*inverse 'space'*	*Line 8465*	*inverse 'D'*
		Line 8470	*inverse 'D'*
		Line 8475	*inverse 'D'*

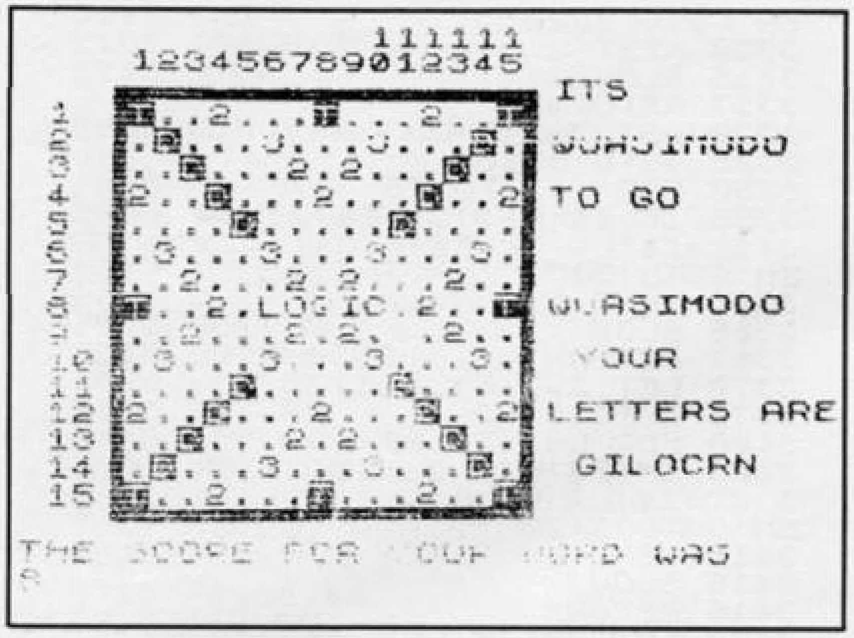
Line by Line

1 to 85	initialisation of variables and instructions.
86 to 170	print the scores, who is next to play, their letters, and inputting coordinates for letters on the board.
171 to 130	work out the coordinates of letters to go on the board.
311 to 318	bonus score of 50 points when a player uses all seven letters in one go.
319 to 502	check for double or triple word score, print out score, and prepare for next player's turn.

Subroutines

3000	checks to see if your letters are next to any others on the board, and adds the necessary score.
3800	POKEs your letter onto the screen.
4000	allocates new letters, and prints them on screen.
5000	Endgame routine. Works out who scored most, and prints out scores.
6000	initialises the array that is to hold the players' letters.
7000	works out if player gets double or triple word-score.
8000	initialises the array that holds the board.
9000	asks how many players there are, asks their names, and DIMs arrays to hold their scores, names and letters.
9500	prints out the board.
9900	routines used when all the letters have been used. Play continues until no-one can go.
9998	program SAVEd so that it autoruns.






```
  1 LET V=0
  2 LET O=0
  3 LET D=0
  4 LET O1=0
  5 LET D1=0
  6 LET W=0
  7 LET Z=0
  8 LET VE=0
  9 RAND
 10 PRINT "SCRABBLE"
 11 PRINT "▀▀▀▀▀▀▀▀"
 12 PRINT
 13 PRINT "(C) R. THOMPSON  18/
4/84"
 14 PAUSE 100
 15 CLS
 19 GOSUB 9000
 20 GOSUB 8000
 22 GOSUB 9500
 25 SLOW
 52 PAUSE 400
 60 CLS
 65 PRINT "TO INPUT YOUR WORD T
YPE IN THE     GRID REFERENCE E.G.
14 FOR         ROW 1,COLUMN 4,104 F
OR             ROW 10,COLUMN 4, OR
1014 FOR       ROW 10,COLUMN 14 AND
 THEN          PRESS NEWLINE"
 67 PRINT
 70 PRINT "THEN TYPE THE REQUIR
ED LETTER      AND PRESS NEWLINE."
 72 PRINT
 73 PRINT "WHEN YOU HAVE FINSHE
D YOUR WORD PRESS NEWLINE AND YO
UR SCORE       WILL BE DISPLAYED."
 74 PRINT
 75 PRINT "TO USE A BLANK (REPR
ESENTED        BY ■) SIMPLY TYPE AN
               INVERSE SPACE."
 77 PAUSE 1100
 78 POKE 16418,0
 79 FAST
 80 CLS
 81 GOSUB 9500
 82 SLOW
 83 PRINT AT 19,0;"A ■ DENOTES
A TRIPLE WORD SCORE A ■ DENOTES
A DOUBLE WORD SCORE A 2 DENOTES
A DOUBLE LETTER     SCORE AND A
3 A TRIPLE LETTER   SCORE"
 85 PAUSE 1000
 86 CLS
 87 GOSUB 9500
 90 FOR I=1 TO P
 91 POKE 16418,0
 92 PRINT AT 19,0
 93 FOR F=1 TO P
 94 PRINT X$(F);"SCORE=";X(F)
 95 NEXT F
 96 POKE 16418,2
 97 PRINT AT 2,20;"ITS";AT 4,20
;X$(I);AT 6,20;"TO GO"
 99 GOSUB 4000
100 FOR T=1 TO 7
101 LET S1=1+PEEK 16396+256*PEE
K 16397
155 LET R=0
157 PRINT AT 20,0;"
          ";AT 21,0;"
           ";AT 19,0;"
158 FOR Q=17 TO 21
159 PRINT AT Q,20;"
160 NEXT Q
164 INPUT L$
165 IF L$="" THEN GOTO 320
170 INPUT M$
171 GOSUB 5500
172 IF R=1 THEN GOTO 155
175 IF LEN L$=2 THEN GOTO 200
180 IF LEN L$=3 THEN GOTO 220
185 LET K=VAL L$( TO 2)
190 LET J=VAL L$(3 TO 4)
191 IF T=1 THEN GOSUB 3000
192 GOSUB 7000
195 GOTO 300
200 LET K=VAL L$(1)
201 IF K<1 THEN GOTO 100
204 LET J=VAL L$(2)
205 IF J<1 THEN GOTO 100
206 IF T=1 THEN GOSUB 3000
207 GOSUB 7000
210 GOTO 300
220 IF VAL L$(1)>1 THEN LET K=V
AL L$(1)
225 IF VAL L$(1)>1 THEN LET J=V
AL L$(2 TO 3)
230 IF VAL L$(2)>5 THEN LET K=V
AL L$( TO 2)
235 IF VAL L$(2)>5 THEN LET J=V
AL L$(3)
240 IF VAL L$(3)>5 THEN LET K=V
AL L$( TO 2)
245 IF VAL L$(3)>5 THEN LET J=V
AL L$(3)
250 IF VAL L$(1)>1 OR VAL L$(2)
>5 OR VAL L$(3)>5 THEN GOTO 285
255 PRINT AT 19,0;"PLEASE BE MO
RE SPECIFIC"
260 PRINT AT 20,0;"WHAT IS THE"
265 PRINT AT 21,0;"ROW NUMBER"
270 INPUT K
275 PRINT AT 21,0;"COLUMN NUMBE
R"
280 INPUT J
285 IF J<1 OR J>15 OR K<1 OR K>
15 THEN GOTO 100
286 IF T=1 THEN GOSUB 3000
290 GOSUB 7000
300 LET B$(K,J)=M$
302 GOSUB 3800
310 NEXT T
311 CLS
313 PRINT AT 10,0;" WELL DONE "
;X$(I)
314 PRINT
315 PRINT " YOUVE USED ALL YOUR
 LETTERS       AND GAINED A BONUS
OF 50          POINTS"
316 PAUSE 200
317 LET Z=Z+50
318 CLS
319 GOSUB 9500
320 IF O1=3 THEN GOSUB 7100
```

```
 321 IF D1=3 THEN GOSUB 7200
 324 PRINT AT 19,0;"THE SCORE FO
R YOUR WORD WAS      ";Z
 325 LET X(I)=X(I)+Z
 326 LET Z=0
 327 PAUSE 150
 328 PRINT AT 19,0;"
                        ";AT 20,0;"
          "
 329 FOR J=1 TO 17
 330 PRINT AT J,20;"
"
 335 NEXT J
 337 IF W=1 THEN RETURN
 340 NEXT I
 345 GOTO 90
 501 GOSUB 9500
 502 GOTO 323
 950 CLS
2999 REM *********************

3000 FAST
3005 IF J=1 AND K=1 THEN GOTO 31
00
3010 IF K=1 AND J=15 THEN GOTO 3
150
3015 IF K=15 AND J=1 THEN GOTO 3
200
3020 IF K=15 AND J=15 THEN GOTO
3250
3025 IF J=1 THEN GOTO 3300
3030 IF K=1 THEN GOTO 3350
3035 IF J=15 THEN GOTO 3400
3040 IF K=15 THEN GOTO 3450
3050 FOR L=1 TO 26
3055 IF B$(K+1,J)=Z$(L) OR B$(K-
1,J)=Z$(L) OR B$(K,J+1)=Z$(L) OR
 B$(K,J-1)=Z$(L) THEN GOTO 3070
3060 NEXT L
3065 SLOW
3067 RETURN
3070 LET Z=Z+F(L)
3075 GOTO 3060
3100 FOR L=1 TO 26
3105 IF B$(K+1,J)=Z$(L) OR B$(K,
J+1)=Z$(L) THEN GOTO 3120
3110 NEXT L
3115 SLOW
3116 RETURN
3120 LET Z=Z+F(L)
3125 GOTO 3110
3150 FOR L=1 TO 26
3155 IF B$(K+1,J)=Z$(L) OR B$(K,
J-1)=Z$(L) THEN GOTO 3170
3160 NEXT L
3165 SLOW
3167 RETURN
3170 LET Z=Z+F(L)
3175 GOTO 3160
3200 FOR L=1 TO 26
3205 IF B$(K-1,J)=Z$(L) OR B$(K,
J+1)=Z$(L) THEN GOTO 3220
3210 NEXT L
3215 SLOW
3217 RETURN
3220 LET Z=Z+F(L)
3225 GOTO 3210
3250 FOR L=1 TO 26
3255 IF B$(K-1,J)=Z$(L) OR B$(K
J-1)=Z$(L) THEN GOTO 3270
3260 NEXT L
3265 SLOW
3267 RETURN
3270 LET Z=Z+F(L)
3275 GOTO 3260
3300 FOR L=1 TO 26
3305 IF B$(K+1,J)=Z$(L) OR B$(K-
1,J)=Z$(L) OR B$(K,J+1)=Z$(L) TH
EN GOTO 3320
3310 NEXT L
3315 SLOW
3317 RETURN
3320 LET Z=Z+F(L)
3325 GOTO 3310
3350 FOR L=1 TO 26
3355 IF B$(K+1,J)=Z$(L) OR B$(K,
J+1)=Z$(L) OR B$(K,J-1)=Z$(L) TH
EN GOTO 3370
3360 NEXT L
3365 SLOW
3367 RETURN
3370 LET Z=Z+F(L)
3375 GOTO 3360
3400 FOR L=1 TO 26
3405 IF B$(K+1,J)=Z$(L) OR B$(K-
1,J)=Z$(L) OR B$(K,J-1)=Z$(L) TH
EN GOTO 3420
3410 NEXT L
3415 SLOW
3416 RETURN
3420 LET Z=Z+F(L)
3425 GOTO 3410
3450 FOR L=1 TO 26
3455 IF B$(K-1,J)=Z$(L) OR B$(K,
J+1)=Z$(L) OR B$(K,J+1)=Z$(L) TH
EN GOTO 3470
3460 NEXT L
3465 SLOW
3467 RETURN
3470 LET Z=Z+F(L)
3475 GOTO 3460
3800 LET V=CODE M$
3805 POKE S1+(K*33+(2*33)+3+J),V
3810 RETURN

4000 REM *********************

4025 IF W=1 THEN RETURN
4030 FOR L=1 TO 7
4035 IF E$(I,L)=" " THEN GOTO 41
00
4040 NEXT L
4045 PRINT AT 10,20;X$(I);AT 12,
20;" YOUR";AT 14,20;"LETTERS ARE
"
4047 PRINT AT 16,21;
4050 FOR L=1 TO 7
4055 PRINT E$(I,L);
4060 NEXT L
4065 PRINT
4070 RETURN
4100 LET VE=VE+1
4101 LET G=INT (RND*100)+1
4103 IF VE>=100 THEN GOTO 9900
4105 IF C$(G)="/" THEN GOTO 4101
4110 LET E$(I,L)=C$(G)
4115 LET C$(G)="/"
4120 GOTO 4040

5000 REM *********************

5001 LET H=0
5003 FOR I=1 TO P
5005 IF H<X(I) THEN LET H=X(I)
5015 NEXT I
5020 FOR I=1 TO P
5025 IF X(I)=H THEN GOTO 5050
5030 NEXT I
5050 PRINT "WELL DONE ";X$(I)
5060 PRINT "YOUVE WON"
5065 PRINT
5070 PRINT "THE SCORES WERE,"
5080 PRINT
5085 FOR I=1 TO 4
5090 PRINT X$(I),X(I)
5095 NEXT I
5100 PAUSE 1000
```

```
5101 CLS
5105 GOSUB 9500
5110 STOP

5500 REM **********************

5600 FOR A=1 TO 7
5605 IF E$(I,A)=M$ THEN LET E$(I
,A)="/"
5607 IF E$(I,A)="/" THEN GOTO 56
20
5610 NEXT A
5615 GOTO 5700
5620 LET E$(I,A)=" "
5625 RETURN
5700 LET R=1
5710 PRINT AT 17,20;"THATS ";AT
18,20;"CHEATING";AT 19,20;"DO IT
";AT 20,20;"AGAIN";AT 21,20;"PRO
PERLEY"
5720 PAUSE 50
5730 RETURN

5999 REM **********************

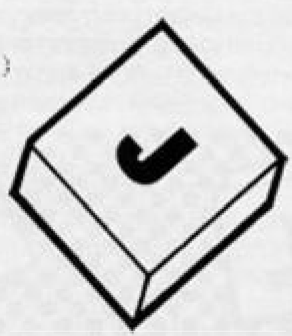
6000 FOR I=1 TO P
6001 FOR M=1 TO 7
6020 LET E$(I,M)=" "
6025 NEXT M
6027 NEXT I
6030 RETURN

6999 REM **********************

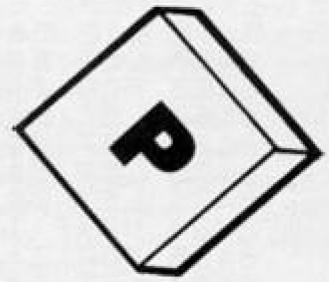
7000 IF B$(K,J)="■" THEN LET Q1=
3
7005 IF B$(K,J)="3" THEN LET Q=3
7010 IF B$(K,J)="■" THEN LET D1=
3
7015 IF B$(K,J)="2" THEN LET D=3
7020 FOR O=1 TO 26
7025 IF A$(O)=M$ THEN GOTO 7040
7030 NEXT O
7035 GOTO 7060
7040 IF Q=3 THEN LET Z=Z+3*F(O)
7045 IF D<>3 AND Q<>3 THEN LET Z
=Z+F(O)
7050 IF D=3 THEN LET Z=Z+2*F(O)
7060 LET Q=0
7065 LET D=0
7070 RETURN
7100 LET Z=Z+(2*Z)
7107 LET Q1=0
7110 RETURN
7200 LET Z=Z+Z
7205 LET D1=0
7210 RETURN

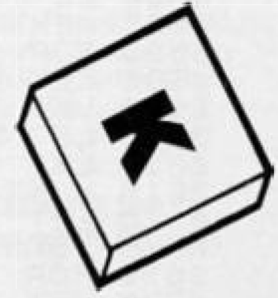
7999 REM **********************

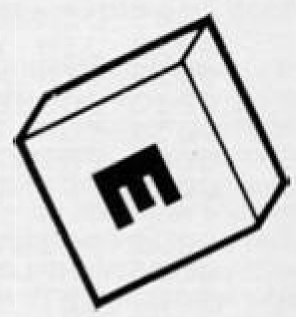
8000 DIM B$(15,15)
8001 DIM A$(26)
8002 DIM F(26)
8005 DIM C$(100)
8007 LET X=0
8010 FOR I=38 TO 63
8011 LET X=X+1
8015 LET F(X)=1
8016 LET A$(X)=CHR$ (I)
8020 NEXT I
8024 LET F(7)=4
8025 LET F(8)=4
8030 LET F(23)=4
8035 LET F(22)=4
8040 LET F(13)=3
8045 LET F(3)=3
8050 LET F(16)=3
8055 LET F(2)=3
```




```
8060 LET F(4)=2
8065 LET F(7)=2
8070 LET F(6)=7
8075 LET F(11)=5
8080 LET F(10)=8
8090 LET F(24)=8
8095 LET F(17)=10
8097 LET F(25)=4
8100 LET F(26)=10
8105 FOR I=1 TO 9
8110 LET C$(I)="A"
8115 NEXT I
8120 LET C$(10)="B"
8125 LET C$(11)="B"
8130 LET C$(12)="C"
8135 LET C$(13)="C"
8140 FOR I=14 TO 17
8145 LET C$(I)="D"
8150 NEXT I
8155 FOR I=18 TO 29
8160 LET C$(I)="E"
8165 NEXT I
8170 LET C$(30)="F"
8175 LET C$(31)="F"
8180 LET C$(32)="G"
8185 LET C$(33)="G"
8190 LET C$(34)="G"
8195 LET C$(35)="H"
8200 LET C$(36)="H"
8205 FOR I=37 TO 45
8210 LET C$(I)="I"
8215 NEXT I
8220 LET C$(46)="J"
8225 LET C$(47)="K"
8230 FOR I=48 TO 51
8235 LET C$(I)="L"
8237 NEXT I
8240 LET C$(52)="M"
8245 LET C$(53)="M"
8250 FOR I=54 TO 59
8255 LET C$(I)="N"
8260 NEXT I
8265 FOR I=60 TO 67
8270 LET C$(I)="O"
8275 NEXT I
8280 LET C$(68)="P"
8285 LET C$(69)="P"
8290 LET C$(70)="Q"
8295 FOR I=71 TO 76
8300 LET C$(I)="R"
8305 NEXT I
8310 FOR I=77 TO 80
8315 LET C$(I)="S"
8320 NEXT I
8325 FOR I=81 TO 86
8330 LET C$(I)="T"
8335 NEXT I
8340 FOR I=87 TO 90
8345 LET C$(I)="U"
8350 NEXT I
8355 LET C$(91)="V"
8360 LET C$(92)="V"
8365 LET C$(93)="W"
8370 LET C$(94)="W"
8375 LET C$(95)="X"
8380 LET C$(96)="Y"
8385 LET C$(97)="Y"
8390 LET C$(98)="Z"
8395 LET C$(99)="■"
8400 LET C$(100)="■"
8402 FOR T=1 TO 15
8405 FOR I=1 TO 15
8410 LET B$(T,I)="."
8415 NEXT I
8420 NEXT T
8425 FOR T=1 TO 15 STEP 7
8430 LET B$(1,T)="■"
8435 LET B$(15,T)="■"
```







```
8440 NEXT T
8445 LET B$(8,1)="▣"
8450 LET B$(8,15)="▣"
8455 FOR I=2 TO 5
8460 LET B$(I,I)="▣"
8465 LET B$(16-I,I)="▣"
8470 LET B$(16-I,16-I)="▣"
8475 LET B$(I,16-I)="▣"
8480 NEXT I
8485 FOR I=2 TO 14 STEP 4
8490 LET B$(6,I)="3"
8495 LET B$(10,I)="3"
8500 NEXT I
8505 LET B$(7,3)="2"
8510 LET B$(7,7)="2"
8515 LET B$(7,9)="2"
8520 LET B$(7,13)="2"
8525 LET B$(8,4)="2"
8530 LET B$(8,8)="*"
8535 LET B$(8,12)="2"
8540 LET B$(9,3)="2"
8545 LET B$(9,7)="2"
8550 LET B$(9,9)="2"
8555 LET B$(9,13)="2"
8560 FOR I=6 TO 10 STEP 4
8565 LET B$(2,I)="3"
8570 LET B$(14,I)="3"
8575 LET B$(I,2)="3"
8580 LET B$(I,14)="3"
8585 NEXT I
8590 FOR I=7 TO 9 STEP 2
8595 LET B$(3,I)="2"
8600 LET B$(13,I)="2"
8605 LET B$(I,3)="2"
8610 LET B$(I,13)="2"
8615 NEXT I
8620 LET B$(4,8)="2"
8625 LET B$(12,8)="2"
8630 FOR I=4 TO 12 STEP 8
8635 LET B$(1,I)="2"
8640 LET B$(15,I)="2"
8645 LET B$(I,1)="2"
8650 LET B$(I,15)="2"
8655 NEXT I
8660 DIM Z$(26)
8663 LET O=0
8665 FOR I=38 TO 63
8667 LET O=O+1
8670 LET Z$(O)=CHR$ (I)
8675 NEXT I
8680 RETURN

8999 REM **********************

9000 PRINT "HOW MANY PLAYERS? (U
P TO FOUR) "
9001 INPUT P
9002 IF P>4 OR P<1 THEN GOTO 900
1
9005 DIM E$(P,7)
9010 DIM X(P)
9012 DIM X$(P,10)
9013 GOSUB 6000
9015 FOR I=1 TO P
9020 PRINT
9021 PRINT "PLAYER ";I;" WHAT IS
"
9022 PRINT "YOUR NAME?";
9025 INPUT N$
9027 LET X$(I)=N$
9030 PRINT N$
9031 PRINT
9035 NEXT I
9040 PRINT "PLEASE WAIT WHILE I
SET UP THE    BOARD"
9050 PAUSE 150
9052 CLS
9055 FAST
9060 RETURN

9499 REM **********************

9500 PRINT AT 0,13;"111111"
9505 PRINT AT 1,4;"1234567890123
45"
9506 LET V=0

9507 PRINT AT 2,4;"▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄
▄▄";AT 2,3;"▗";AT 2,19;"▖"
9510 FOR C=1 TO 15
9511 LET V=V+1
9515 PRINT AT V+2,1;C;AT V+2,3;"
▐";
9517 PRINT B$(C);"▌"
9520 NEXT C
9530 PRINT "    ▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀"
9535 PRINT AT 18,3;"▝";AT 18,19;
"▘"

9540 RETURN

9899 REM **********************

9900 CLS
9901 PRINT AT 10,0;"THERE ARE NO
 MORE LETTERS LEFT"
9904 PRINT "YOU MUST PLAY ON UNT
IL NONE OF  YOU ";P;" CAN MAKE A
NY MORE WORDS"
9905 PAUSE 200
9906 CLS
9907 PRINT AT 10,0;"CAN ANY OF Y
OU GO?(Y/N)"
9908 INPUT V$
9909 IF V$="N" THEN GOTO 9980
9910 CLS
9911 GOSUB 9500
9912 FOR I=1 TO P
9917 FOR G=1 TO 11
9918 PRINT AT G,20;"
"
9920 NEXT G
9922 PRINT AT 1,20;X$(I);AT 3,20
;"YOUR LETTERS ";AT 5,20;"ARE";A
T 7,20;
9925 FOR A=1 TO 7
9926 PRINT E$(I,A);
9927 NEXT A
9930 PRINT AT 9,20;"CAN YOU ";AT
 11,20;"GO (Y/N)?"
9933 INPUT S$
9935 IF S$="N" THEN GOTO 9940
9937 LET W=1
9938 IF S$="Y" THEN GOSUB 99
9940 NEXT I

9970 GOTO 9906

9980 CLS
9981 GOTO 5000
9997 STOP
9998 SAVE "SCRABBL▣"
9999 RUN
```



ZX81 Soft Selection

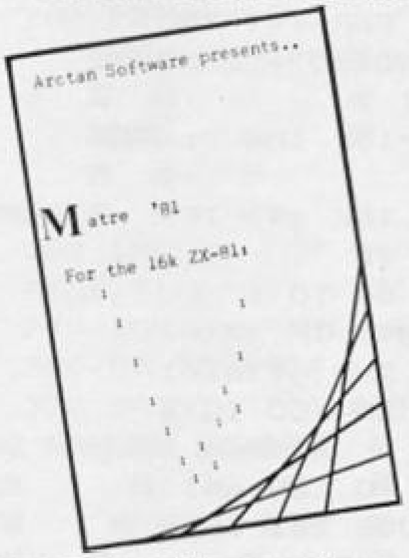
Nick Pearce is back with some more new '81 Software.

Matre 81
Arctan Software

Matre 81 is an interesting graphics program with some unusual functions for the 16K ZX81. Probably the most unique is 'kaleidoscope' where anything drawn in the top right hand quadrant of the screen is mirrored in the other three quadrants.

There are some thirty functions in all. To start, you are presented with a blank screen and asked how many pictures you wish to draw — up to eight separate pictures can be handled by the computer.

In Draw mode you move the



cursor around the screen to produce the required pattern or picture. You can move horizontally, vertically, and diagonally (eight directions in all). The drawing area has 43 vertical and 64 horizontal pixels, so picture resolution is not all that good.

Function mode can be called up at any time. Each function is called by a single character, and you are asked to confirm that any selected function is indeed required before it is operated. This slows things up a little, but does reduce the probability of error. The functions available include the usual Copy, Save, and Load routines as well as such features as Invert, Fill (to fill the screen with a specified character), Charborder (to get a border using your own choice of character), Store and Recall (to get pictures into and out of memory) and Cursoring (which moves the cursor directly to a specified corner of the screen).

There is an animate function which displays each picture consecutively. However as there is a maximum of eight pictures, animation is strictly limited.

All the features worked as indicated on the comprehensive sheet that is included with the cassette. The program is entertaining and is very easy to use. I am afraid that I am unable to illustrate this review with any of my doodles as my Sinclair printer has thrown a wobbly — well I did drop it. Is there anyone out there who can repair the little beasties?

Matre 81 costs £3.75 (cheque payable to MG Huggins) from Arctan Software, Suite 2, Assurance House, 35-41 Hazlewood Road, Northampton NN1 1LG.

Pooter Puzzler and Muse of Pooter
Pooter Games

Pooter Puzzler contains a collection of four word games, all for the 16K ZX81.

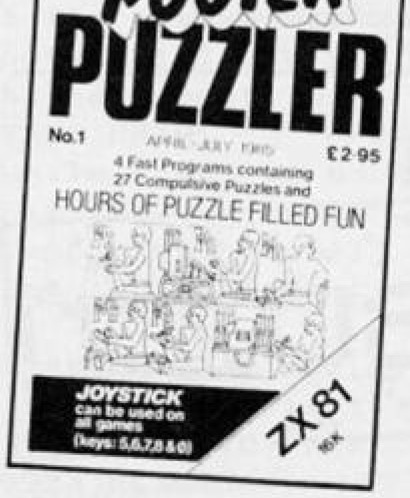
The first is called Sortition. The computer holds 26 nine letter words, each starting with a different letter of the alphabet. The player selects a letter from A to Z, and the computer displays an anagram of the stored word beginning with that letter. Your task is to solve the anagram, and sort the letters into their correct order. Sorting is not easy since it has to be done by progressively reversing the order of the displayed letters — from the left only. A score is kept (number of moves taken) which the computer averages out as more words are attempted. There is also a two player option.

Sortition is quite a good game, but can be rather frustrating at times particularly as there is no way of giving in and requesting the answer — although you can escape by pressing the break key.

The next game on the cassette is Switch-Quiz, which I thought was jolly good fun. It is a quiz where the answer to each question is either true or false. You get points for giving the correct response, and the faster you answer the more points you get. For an incorrect answer points are lost. To start you make a selection from the nine topics available, from pop history and film buff, to sport and British birds. It is then up to you to give the correct response to each question as quickly as possible and reach the maximum permitted score of 90 points in the shortest possible time. The questions have been carefully thought out and it is easy to give a wrong answer under the pressure of time. Good use has been made of ZX81 graphics with a points scoring clock displayed throughout the game.

Next comes Wordslide, containing eight puzzles. Each puzzle is a four by four grid containing four words with their letters jumbled up. You have to move the letters around the grid to recreate the four words. It is a ZX81 version of the popular novelty puzzle. An enjoyable game with graphics again used to good effect.

Finally on this cassette is Word Scanner. Thirty or more words are hidden in a 16 by 16 grid. They may be horizontal, vertical, or along a diagonal. The object of course is to find each word. Ten puzzles are available, each based on a theme such as top thirty, British mountains, and garden roses. The game is easy to use by simply moving a cursor around the grid and 'firing' to lock on to the located word. A list of the hidden words is displayed.



All four games on this cassette worked well. They have obviously been carefully thought out and were enjoyable to play. They can all be played using a joystick.

Also from Pooter Games is an interesting novelty for the expanded ZX81 called *Muse of Pooter*, a program which creates prose. Short sentences are generated and displayed, with words and syntax chosen at random.

Most of the phrases it generated whilst I used the program seemed plausible, even profound at first sight, but on closer inspection probably 90% in fact made no sense. Occasionally though, a real beauty appears — so switch on and let the Muse of Pooter set you pondering.

Pooter Puzzles and Muse of Pooter are available by mail order from Pooter Games, 24 Parsloes Avenue, Dagenham, RM9 5NX.

Bouncing Bert
Software Farm

The last Software Farm game for the ZX81 reviewed in these pages featured high resolution graphics. Normal ZX81 graphics for *Bouncing Bert*, but the display and animation are very good nevertheless. The program requires 16K of RAM.

Bouncing Bert is not the usual pinman, but a substantial character who can move forwards, backwards, and can jump. The object of the game is to guide him from platform to platform to collect some keys and return them to the start. There are three keys to collect in all, and there are plenty of hazards to trap the intrepid Bert, including weeds and deathly skulls.

An interesting aspect to this game is that everything from Bert to the skulls and weeds is on a large scale. This means that only a small part of the game can be seen on screen at any time. This makes it difficult to plan a route to the keys and only after a number of attempts after which the course is "learnt" is success likely.

Bouncing Bert is an enjoyable game, and good use is made of the ZX81 graphics for the display and animation. Available from Software Farm, 3rd Floor, 16 Charlotte St, Bristol BS1 5PP. ■

BUZZ!

Derek Mearns has been a busy little B — he can be seen buzzing around the Witney Honey pots.

In this game for the 16K ZX Spectrum you must manoeuvre the fly around the orchard to the fence on the other side. It differs from the normal arcade games in that your movements are via an input statement at the beginning rather than by INKEY$ during play.

When run you will see a short title display followed by the difficulty input. This determines how many apples you will get (eighteen times the difficulty level). The screen display will then appear, with you at the bottom and the fence at the top. You will then see the prompt 'O.K!' indicating that you should now enter your commands using the cursor keys. All the commands should be entered at the beginning. For example, if there was a straight path to the other side except for one apple blocking your way, then your input may be:

'77777777577877777777'

Notice that it is not necessary to finish at the fence, you may over-shoot it. However, you may not under-shoot it. Should the maze be impossible entering 0 will return you to the start.

Program Description

Here follows a brief breakdown of the program and a list of the variables used. (See figure 1)

Figure 1. Program Description)

10-38	Set up User Defined Graphics and input difficulty
39-50	Draw screen display
60-70	Input commands
80-260	Move fly and check to see if at fence or crashed
280-290	The fly crashes
320-360	The fly reaches the fence
400	End of command and fly not at fence
1000-1020	Another go?
9000-9520	Print title and instructions
dif	Difficulty level
x,y	Co-ordinates of the fly
g$	Your command
n	Next command to be read in g$ .FOR. . . NEXT loops
a$	Temporary scrolling messages
f,g	FOR. . . NEXT loops

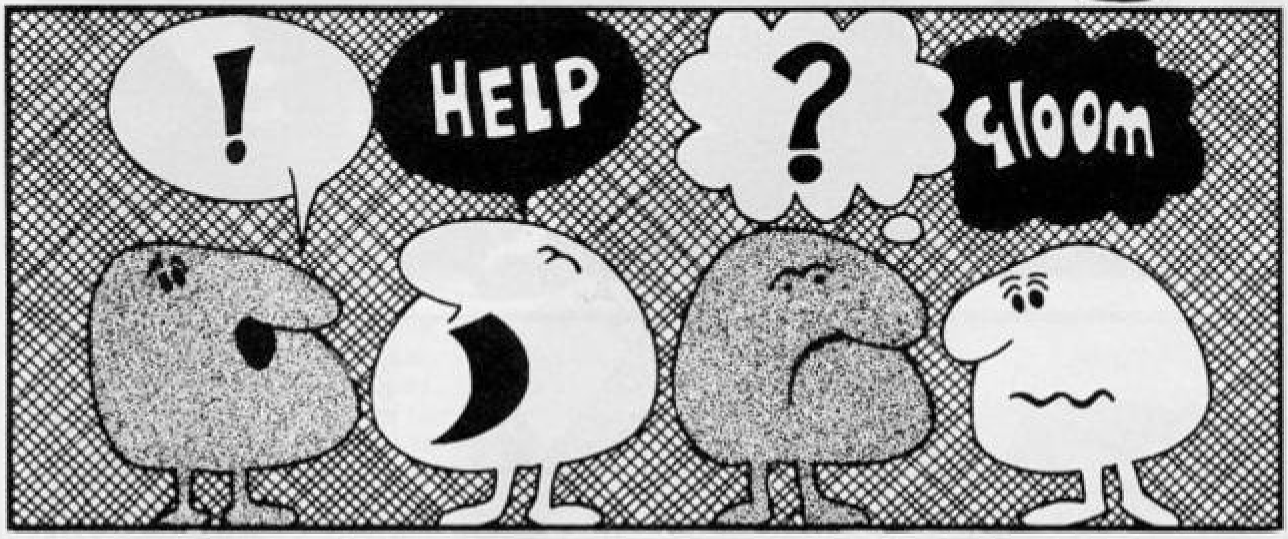



```
   1 REM ***********************
         *Underlined characters*
         *are entered in       *
         *GRAPHICS mode.       *
         ***********************
   5 GO SUB 9000
  10 RESTORE
  20 FOR g=144 TO 147: FOR f=0 T
O 7: READ a: POKE USR CHR$ g+f,a
: NEXT f: NEXT g
  30 DATA 0,6,8,28,62,62,62,28:
DATA 36,24,60,90,153,153,165,195
  33 DATA 240,136,69,62,62,69,13
6,240: DATA 8,255,8,255,0,0,0,0
  35 INK 7: PRINT AT 8,3;"Enter
Difficulty (1 TO 10)": INPUT dif
: IF dif<1 OR dif>10 THEN  GO TO
 35
  37 LET dif=INT dif: LET dif=di
f*18: PRINT AT 8,3;"
                 ";AT 1,1;"
                          "
  38 PRINT AT 9,1;"
                         "
  39 INK 6: PRINT AT 0,1;"DDDDDD
DDDDDDDDDDDDDDDDDDDDDDDD"
  40 PAPER 0: INK 4: FOR f=1 TO
dif: PRINT AT INT (RND*15)+1,INT
 (RND*30)+1;"A": NEXT f
  50 LET x=16: LET y=15: INK 7:
PRINT AT 16,14;" B "
  60 INPUT "O.K! "; LINE g$: IF
LEN g$<1 THEN  GO TO 60
  65 IF g$="0" THEN  GO TO 1
  70 FOR n=1 TO LEN g$: IF g$(n)
<>"5" AND g$(n)<>"6" AND g$(n)<>
"7" AND g$(n)<>"8" THEN  GO TO 6
0
  80 NEXT n: FOR n=1 TO LEN g$:
PRINT AT x,y;"B": PAUSE 20: IF g
$(n)="5" THEN  GO TO 120
  90 IF g$(n)="6" THEN  GO TO 16
0
 100 IF g$(n)="7" THEN  GO TO 20
0
 110 IF g$(n)="8" THEN  GO TO 24
0
 120 INK 7: IF ATTR (x,y-1)=2 TH
EN  GO SUB 280: PRINT AT 19,0; F
LASH 1;" YOU FLEW INTO A WALL AN
D DIED! ": GO TO 300
 130 IF ATTR (x,y-1)=4 THEN  GO
SUB 280: PRINT AT 19,0; FLASH 1;
"YOU ATE A DEADLY APPLE AND DIED
!": PAUSE 200: GO TO 300
 140 PRINT AT x,y;" ": LET y=y-1
```

```
: BEEP .2,10: NEXT n: GO TO 400
 160 IF x>14 THEN  GO SUB 280: P
RINT AT 19,0; FLASH 1;" YOU FLEW
 INTO A WALL AND DIED! ": PAUSE
200: GO TO 300
 170 IF ATTR (x+1,y)=4 THEN  GO
SUB 280: PRINT AT 19,0; FLASH 1;
"YOU ATE A DEADLY APPLE AND DIED
!": PAUSE 200: GO TO 300
 180 PRINT AT x,y;" ": LET x=x+1
: BEEP .2,10: NEXT n: GO TO 400
 200 IF x<2 THEN  GO TO 320
 210 IF ATTR (x-1,y)=4 THEN  GO
SUB 280: PRINT AT 19,0; FLASH 1;
"YOU ATE A DEADLY APPLE AND DIED
!": PAUSE 200: GO TO 300
 220 PRINT AT x,y;" ": LET x=x-1
: BEEP .2,10: NEXT n: GO TO 400
 240 IF ATTR (x,y+1)=2 THEN  GO
SUB 280: PRINT AT 19,0; FLASH 1;
" YOU FLEW INTO A WALL AND DIED!
 ": PAUSE 200: GO TO 300
 250 IF ATTR (x,y+1)=4 THEN  GO
SUB 280: PRINT AT 19,0; FLASH 1;
"YOU ATE A DEADLY APPLE AND DIED
!": PAUSE 200: GO TO 300
 260 PRINT AT x,y;" ": LET y=y+1
: BEEP .2,10: NEXT n: GO TO 400
 280 PRINT AT 6,0;"      ███ ███ █
    ███ ███  █   █   █ █ █
    █ █  █   █   ███ ███ █
    ███  █   █    ██ █   █
    █ █  █   █   ███ █   █
███ █ █  █   █  █ "
 285 INK 2: FOR f=6 TO 10: PRINT
 AT f,0;"█";AT f,31;"█": NEXT f
 290 INK 0: PAPER 3: RETURN
 300 GO TO 1000
 320 PRINT AT 6,0;"      ███  █
█ ███ ███ █ █ █ █
█   █   █ █ ███ █
█  █   █  █ █ █ █
█ █   █   █ █ █ █
█ █  █      ███ ███
   ███ ███ █ "
 330 INK 2: FOR f=6 TO 11: PRINT
 AT f,0;"█";AT f,31;"█": NEXT f
 340 FOR g=1 TO 2: FOR f=20 TO 4
0 STEP 3: BEEP .1,f: NEXT f: NEX
T g
 345 LET a$="!!!!!WELL DONE,YOU
MADE IT!!!!!!"
 350 PRINT AT 19,1; INK 2;"BBBBB
BBBBBBBBBBBBBBBBBBBBBBBBBB": FOR
 f=1 TO 30: PRINT AT 19,f; INK 7
;a$(f);"C": PAUSE 15: NEXT f
 360 GO TO 1000
 400 GO SUB 280: PRINT AT 19,0;
FLASH 1;"      YOU DID'NT MAKE IT
 !      "
1000 FOR f=1 TO 500: NEXT f
1010 INK 6: PAPER 0: PRINT AT 19
,0;"Press any key to play again
    ": IF INKEY$="" THEN  GO TO
1010
1020 RUN
9000 PAPER 0: BORDER 0: CLS
9003 BORDER 0: PAPER 0: CLS : IN
K 2: FOR f=0 TO 3: PLOT f,40: DR
AW 0,135: PLOT 0,175-f: DRAW 255
,0: PLOT 255-f,175: DRAW 0,-135:
 PLOT 255,40+f: DRAW -255,0: NEX
T f
9004 FOR f=1 TO 4
9005 INK 6
9010 LET a$="BUZZZ!************
**********": FOR n=31 TO 1 STEP
 -1: PRINT AT 1,n;a$( TO 31-n):
NEXT n
9015 FOR n=1 TO 7: PRINT AT 1,1;
a$(n TO ): NEXT n
9017 LET a$="  DEREK MEA
RNS  ":
9020 INK 5: FOR n=30 TO 1 STEP -
1: PRINT AT 1,n;a$(n): NEXT n
9025 NEXT f
9100 RETURN
9200 CLS : INK 7: PRINT AT 0,0;"
  The object of this game is to
guide Fred the Fly(B) around the
orchard.Unfortunately Farmer
Brown doesn't like flies and he
has poisoned all the apples in
his orchard.Fred will die unless
";
9210 PRINT AT 6,0;"you can guide
 him into the next orchard where
 the apples are notpoisoned.The
only way into this orchard is th
rough the fence at the top(D)"
9220 PRINT AT 11,0;"  To guide
Fred use the cursor keys BUT you
 must enter your    route accros
s the orhard all at once.The com
puter will ask you  for this wit
h the prompt 'O.K'  "
9230 PRINT "   Pressing any key
while Fred  is moving will make
him fly     faster.Should it be
impossible  to fly through enter
ing '0' willstart the game again
."
9240 PRINT AT 21,0; INK 6;"Press
 any key to start": IF INKEY$=""
 THEN  GO TO 9240
9250 RETURN
```



Problem Page



Chess

Dear Sir,
Some time ago I bought a 1K Chess program for my ZX81. After trying several times it seemed impossible to BREAK the program. Do you know how this is done because I have a program in my head which I don't want to be broken.

Th.P. Van Nunspeet
The Netherlands

Well, presuming it is the program and not your head which you want to prevent from being broken, then this can be a simple problem providing your program is in machine code. I too have this chess program and it is a masterpiece of programming, the code is held in a REM line at the beginning of the program and the BASIC simply consists of two lines.

```
10 SAVE "CHESS"
20 RAND USR number
```

So, if your program is in machine code held in a first line REM just add two similar lines to the end of your program remembering to put 'number' equal to the machine code call address (often 16514), of the start of your program.

Now save your program by typing GO TO 10 and pressing NEWLINE. Every time the program is reloaded from the tape it will auto start and, provided your machine code does not allow BREAK to halt the program it will be protected.

Saving a program in this way will also autostart a BASIC program if the RAND USR number is replaced by GOTO line number, but preventing a BASIC BREAK is too complex to tackle here.

Eraser

Dear Sir,
I have been looking for a way to erase a BASIC program if the program is stopped while loading so revealing the structure of the program and also making it easier to be copied.

I have found the POKE commands to erase the program if the BREAK/SPACE key is pressed when the program has loaded. I wish to find a POKE command to erase the program if the program is stopped while loading. Can you help me? I own a 48K Spectrum and thoroughly enjoy your magazine.

Michael Conaty,
Dublin

This seems to be of interest at the moment. There really is no foolproof way of protecting a program, just look at the time and effort commercial companies take, and still the hackers get in! Without recourse to machine code and one of two other tricks, the simplest way is to have one or two pre-loaders, each depending on a secret code you implant for the next to work. The main program could be saved as bytes, loaded to a different memory location and therefore only be reloaded by the correct pre-loader.

Say for instance you have a BASIC program. SAVE it as bytes by having a program line such as:

```
9990 SAVE "prog" CODE
23552, (PEEK23641 + 256 *
PEEK23642) - 23552 9995
GO TO 10
```

*Now if it is reloaded using LOAD "" CODE, then it will load to the original address and continue from where it left off, this we don't want — yet. First find the length of the program saved with PRINT (PEEK 23641 + 256 *PEEK 23642) – 23552 and make a not of it. Make a pre-loader by entering a program such as:*

```
10 POKE 23659,0:POKE
65365,0:REM cause crash on
break
20 LOAD "" CODE 23552
```

By clever use of colour, line 20 can be in white ink on white paper ie. invisible! Or, by a similar technique made to look as if it reads LOAD "" or even LOAD "" CODE. It all adds to the hackers' confusion.

Save it by SAVE "p1" LINE 10. Now reload the first program to a different address by CLEAR 32999 (or some such address) and LOAD "" 33000. Re-save by SAVE "prog" CODE 33000, length of prog as noted, after your pre-loader program. Now a simple LOAD""CODE will be of no use!

Saving from different areas of the System variables helps to prevent others getting used to the technique. There are lots of possible tricks which can be used, but none are foolproof, all you can really do is make life difficult and hope the would-be hacker gives up.

Convertable?

Dear Sir,
Having used a ZX81 for years, I have acquired a Spectrum. Although I intend to keep the ZX81 and keyboard as a backup, life would be much easier if I could load my BASIC ZX81 tapes into the Spectrum — to avoid the need to type them all out again.

Do you Know of any conversion tape still available? I have tried ELR, Amersham and Ness.

Mr. M. Anthony,
Cheshire

The answer is simple — no! The ELR one is the only one we knew of, but this one only worked with issue 1 Spectrums. Perhaps if there is such a product in existence someone will let us know about it. Meanwhile it might be an interesting project for an enterprising expert out there . . . . . .

David has dealt with the bulk of the problems personally this issue and we were left with only these few. However, I will be looking after this page for a while to allow David to work on his new project so send in your program, hardware or general problems and queries to me:

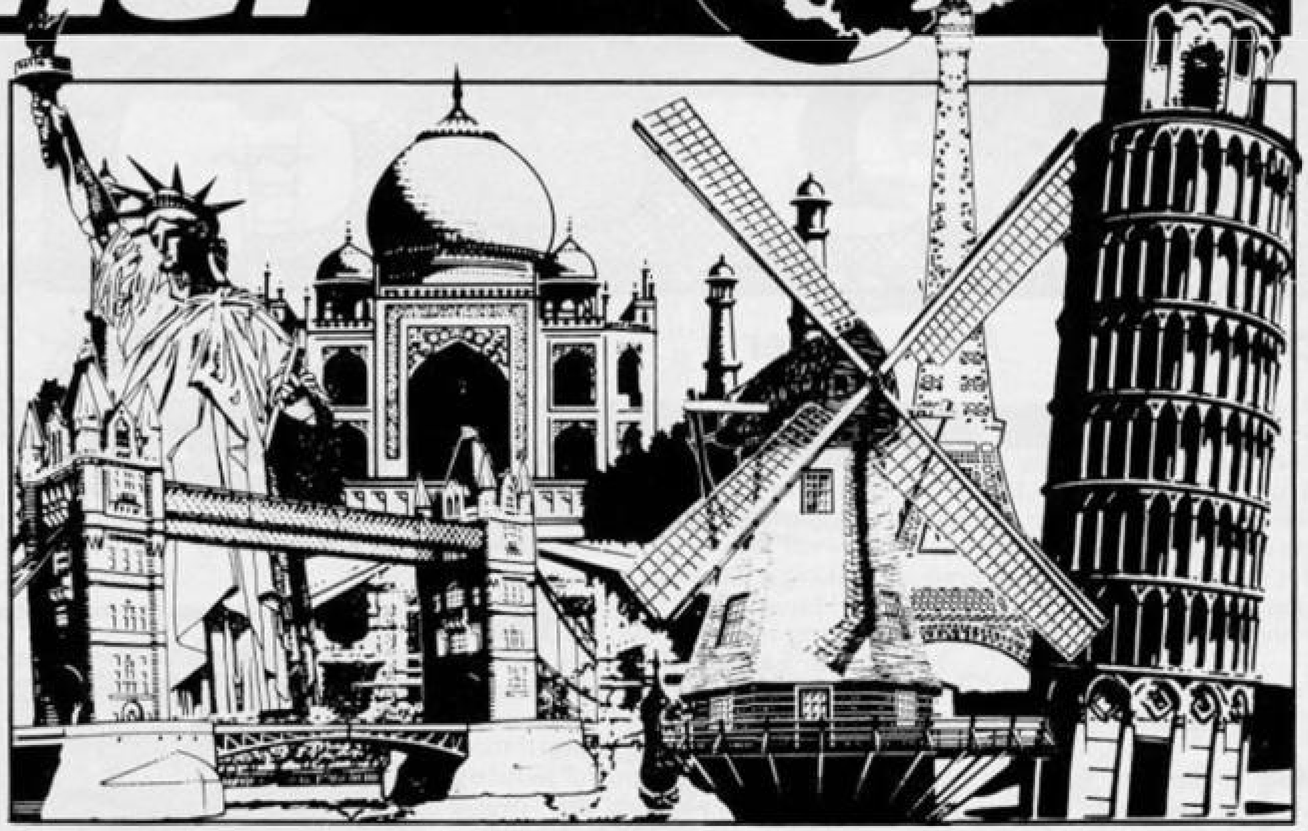
Sparky,
Problem Page,
ZX Computing,
No 1 Golden Square,
London W1R 3AB

And I look forward to a challenging and interesting time.



Club corner





ZX81 club

Dear Editor,
In March of this year one of your rival magazines printed an appeal, by myself, for ZX81 owners and users to contact me if they were interested in establishing a ZX81 postal users club. Over two dozen people replied (including one each from Canada, Germany, New Zealand and Sweden). This excellent response has resulted in tentative steps being taken to form such a club.

Facilities and services which we hope to be able to provide include: address exchange, discount software (over 50 titles at less than £3.00 each!), International and Junior sections, newsletter etc.

Therefore, I would like to ask, through your columns, that ALL ZX81 owners and users, who would be interested in joining a club EXCLUSIVELY for themselves, please write to me at the above address, for further information and a questionnaire. Would all respondents please be kind enough to enclose a stamped addressed envelope or an international reply coupon. Many thanks!

Yours sincerely
Iain Dale
78 Cobden St,
Thornaby, Stockton on Tees,
Cleveland TS17 7ET

Strathclyde club

Dear ZX Computing,
I am starting a new, nationwide Sinclair User Club. Each week I will send members a newsletter which will consist of games reviews, tips, short routines, discount computer games and more.

The cost of membership will be:
Spectrum — £10 per year or £1 per month.
ZX81 — £5 per year or £0.50 per month.

Your money will contribute towards postage, discount games tapes and to any other surprises that may appear in the newsletter. I think you will find the lub good value for money, but if you don't enjoy it you will have a money back guarantee for the first two months.

If you are interested please contact:

Ian Kennedy
24 Waverley Crescent
Lanark, Scotland.

Spain

Hello friends of ZX Computing
I am the president of the programmers club of Castellon (Spain), and I would like to know people from all over the world who own Sinclair, Commodore or Astra computers.

We (the members of the club) want to exchange ideas, programs etc without any financial interest involved.

Our address is:

Jose Manual Martin Sautos
Spdo 635, Castellon
Spain.

Penpal

Dear ZX Computing
I am fourteen years old, and would like a penpal who has a Spectrum 48K. I have an Interface 2 and joystick for my Spectrum 48K.

Yours faithfully
G. Bentham
PO Box 73
Ngodwana, N.E. Transvaal
RSA 1209

While we're on the subject of S. Africa, we've had a letter from a Peter Hupton, and also from the 'ZX Mushroom Club', both of whom are looking for penpals/new members but didn't send in their full addresses. So if they're reading this and would like to write again with their addresses, we'd be happy to publish their letters.

France

Dear ZX Computing
I would be very pleased if you would publish this letter.

I am French and live near Paris, and I own a 48K Spectrum. I would like to become a member of a Spectrum club as I would like to exchange ideas, news and programs with British users.

If there are any clubs that correspond with French or European users, please write to me at the address below.

Yours faithfully
Pajard Jerome
57 Rue Segoffin
92400 Camberoie
France

Eire

Dear ZXC
I would like to tell any of your North Kildare readers that I hope to form a small club of Spectrum owners. Anybody in the area who is interested may write to me, enclosing s.a.e., or phone after 7pm at (045) 68144.

Yours sincerely
Owen O'Connor
Downings North
Prosperous
Co. Kildare, Eire

De-bugger

Getting a program typed in is often only the start of your problems. Ed to the rescue.

Typing in a program is a useful exercise. Apart from the patience required, techniques learned and the end program to be used, probably the most educational part of it is tracking down the bugs introduced by yourself or occasionally by our publication system.

In debugging you gain a much deeper insight and understanding on how the program actually works than by merely typing it in, but tracking down these errors is an art in itself and needs some skill. So here are some tips to help you in your efforts when faced with that cryptic error report!



1 NEXT without FOR
Look back through the program, either the loop has not been set up — no related FOR 'letter' = No1 TO No2 line, or the letter has been re-used as an ordinary variable within the loop with a LET 'letter' = No.

2 Variable not found
This is one of the most common errors. Again, the problem may not lie in the line where the error was detected and reported. If there is only one variable, which may be one or more letters or a string ($) variable, then that is the problem. There may be more than one variable in the line section reported and you will have to identify the offending one. In a line PRINT AT Y,X;A$ the culprit could be Y or X or A$. To find out which of them is causing the problem (it may be more than one) type in turn as a direct command:

```
PRINT Y Enter/Newline
PRINT X Enter/Newline
PRINT A$ Enter/Newline
```

Note which produces the error report. Now look back through the program printout for the line which sets it up — usually a LET or FOR command. Did you leave it out? Does the program get there or has a GOTO/GOSUB been wrongly addressed?

3 Subscript wrong
Connected with DIM A(No) or DIM A$(No). If the number in the brackets on the line where the error is reported is greater than the one in the original DIM statement, is not an integer or is less than 1, then this report is generated. If the subscript — number in brackets — is a number then check and change, however, if it is a variable then follow the procedure for tracing variables. It has probably exceeded the limits, look for lines with the variable being altered with + − ✱ / : if necessary add limiting code. For example:

```
IF X >10 THEN LET X = 10
```

4 Out of memory
As well as for programs which are too big, it may happen if the previous program set RAMtop. Before despairing, enter CLEAR USR "a"-1 on the Spectrum: on the ZX81 SAVE the program, turn the machine off and on, then reload the program.

7 RETURN without GOSUB
Somehow the computer has reached a RETURN command other than via a GOSUB instruction. Check a GOTO hasn't been entered in place of a GOSUB. Check for a missing GOSUB.

B Integer out of range
An integer (whole number) either as a number of variable is too big or small and you are attempting to do something like PRINT AT 0,33 — not allowed! Check any variables involved as per report 2 and trace it back through the program looking for adjustments to it by + − ✱ / : Add limiting code if needed — see report 3

E Out of DATA
A Spectrum problem. Check the number of DATA items match the number of READs; usually one (or more) has been missed out. Attempting to reread a DATA list without first using a RESTORE command will cause this and it can happen on an auto start program (saved with a LINE number). Good programming usually RESTOREs to the correct line number before using READ.

I FOR without NEXT
See report 1 but this time the NEXT is missing!

Note that the letters I have used for examples could by ANY letters not just A$,X,Y etc and depend on the particular choice of the programmer.

This is by no means a comprehensive list but I have tried to cover many of the most common error reports. Personally, I get almost as much satisfaction from debugging as I do from programming I do assure you, however, that there is absolutely no truth in the rumour that we deliberately inject bugs into our listings in order to introduce you to the dubious delights of debugging! ■

Conversion tips

A guide to ZX81/Spectrum program conversions from David Nowotnik.

The versions of BASIC offered by the two ZX computers are so similar that many programs for one can be used by the other. The ZX81 has only two commands which are not present on the Spectrum, SCROLL and UNPLOT, and these should cause you few problems when converting ZX81 programs to the Spectrum (see Table 1).

There are quite a lot of commands and functions on the Spectrum which are not available on the ZX81. A list of these appears in Table 4. The stars indicate those commands and functions for which there is no simple translation to ZX81 BASIC. Those for colour and sound can be ommitted; but you will have to find some alternative for the high resolution and file I/O commands.

The command PLOT appears on both computers, but the effect is quite different, so beware! Another tip: PEEK and POKE should be used with caution. In conversion, addresses will almost certainly have to be changed. Some of those changes appear in the tables. A command such as POKE USR "a" . . . . on the Spectrum indicates User Defined Graphics; ZX81 users don't have this facility, so you'll have to omit this and use a standard character instead.

ZX81	Spectrum	Comments
SCROLL	RANDOMISE USR 3582 or LET t=USR 3582	If the program uses random numbers, they could become rather predictable with the first option. If so, use the second, using a variable (in this case t) which is otherwise not used.
PLOT Y,X	PRINT AT 21−Y/2,X/2;	Print the appropriate quarter square graphics character.
UNPLOT Y,X	PRINT AT 21−Y/2,X/2;	Print a space, or the appropriate quarter square graphics character.

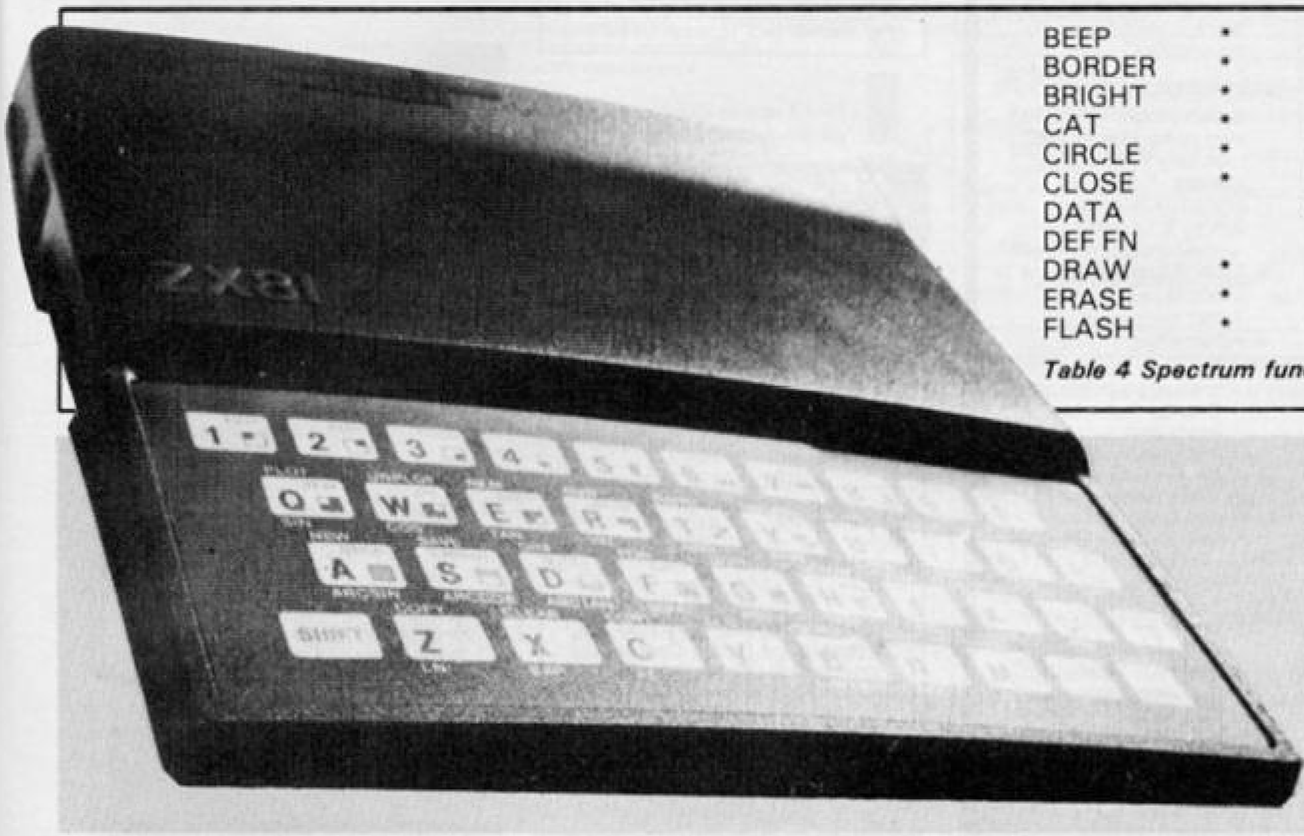
Table 1 ZX81 to Spectrum conversions.

Spectrum	ZX81	Comments
BIN eg LET y=BIN 10010101	LET y=(decimal no.) Conversion to decimal: 10010101=149 128 64 32 16 8 4 2 1 Add these numbers together when a 1 appears at the appropriate position in binary.	BIN allows the representation of a number in binary. On the ZX81 use the decimal equivalent, but beware; BIN is often used with User Defined Graphics, which are not available on the ZX81.
READ/DATA eg READ x,y DATA 50,60	LET LET X=50 LET Y=60	READ and DATA are used to store a lot of information in a program. Use LET instead.
DEF FN and FN eg DEF a(x)=SQR x LET t=FN a(i)	LET X$="SQR X" LET X=I LET T= VAL X$	The defined function can appear in a string. Use the keyword for built-in functions (eg SQR). The equivalent of FN may need 2 lines, as shown.
PLOT	no equivalent	
SCREEN$ eg LET a=SCREEN$ x,y	LET A=PEEK(PEEK 16396 +256*PEEK 16397+1+Y+33*X)	Used in interactive games to detect characters in the display file. Note — this formula only works when a RAM pack is fitted.

Table 2 Spectrum to ZX81 conversions.

ZX81	Spectrum	Comments
1 FRAMES POKE 16436,255 POKE 16437,255	POKE 23672,0:POKE 23673,0	Both computers have a counter which accurately varies by 50 every second. In the example, use the first line to start the 'clock'. The variable T will have the time in seconds after the start. The counter can only be used for 10 minutes.
LET T=(65535–PEEK 16436–256*PEEK 16437) /50	LET t=(PEEK 23672+256* PEEK 23673)/50 For times greater than 10 minutes, you can use byte 23674 as well.	
2 Line number zero POKE 16510,0	POKE 23756,0 (As the start of BASIC can move, eg with microdrives) use with caution.	Converts the first line of a program to line number zero, which cannot be edited, and so is protected.
3 RAMTOP POKE 16388,X–256*INT (X/256) POKE 16389, INT (X/256)	CLEAR x	Creates a safe area at the top of RAM starting at address x, for storing data, machine code etc.

Table 3 General interconversion hints.



BEEP	•	FORMAT	•	ATTR	•
BORDER	•	INK	•	BIN	
BRIGHT	•	INVERSE		FN	
CAT	•	MERGE	•	IN	•
CIRCLE	•	MOVE	•	OVER	•
CLOSE	•	OPEN	•	POINT	•
DATA		OUT	•	SCREEN$	
DEF FN		PAPER	•	VAL$	•
DRAW	•	READ			
ERASE	•	RESTORE	•		
FLASH	•	VERIFY	•		

Table 4 Spectrum functions not available on the ZX81.

System Variables Conversion Table.

Variable	ZX81/ T/S1000	Spectrum/ TS2068
BREG	16414	23655
CDFLAG	16443	No Equivalent
CH ADD	16406	23645
COORDS	16438	23677
COORDS (Byte 2)	16439	23678
DEST	16402	23629
DF CC	16398	23684
D FILE	16396	No Equivalent
DF SZ	16418	23659
E LINE	16404	23641
ERR NR	16384	23610
E PPC	16294	23625
ERR SP	16386	23613
FLAGS	16385	23611
FLAGX	16429	23665
FRAMES	16436	23672
LAST K	16421	23560
MARGIN	16424	No Equivalent
MEM	16415	23656
MEMBOTT	16477	23698
MODE	16390	23617
NXTLIN	16425	23637
OLDPCC	16427	23662
PPC	16391	23621
PRBUFF	16444	23296
PR CC	16440	23680
RAMTOP	16388	23730
SEED	16434	23670
S PSN	16441	23688
S POSN (Byte 2)	16442	23689
STKBOT	16410	23651
STKEND	16412	23653
S TOP	16419	23660
STRLEN	16430	23666
T-ADDR	16432	23668
VARS	16400	23627
VERSN	16393	No Equivalent
X PTR	16408	23647

ZX COMPUTING

Lineage: 40p per word.

Semi display: £9.00 per single column centimetre. Ring for information on series bookings/discounts.

All advertisements in this section must be prepaid.
Advertisements are accepted subject to the terms and conditions printed on the advertisement rate card (available on request).

01 - 437 0699

Send your requirements to:
DAVID HORNE
ASP Ltd., 1 Golden Square, London W1

SOFTWARE

CAR CURE



★ **REVIEWS**

"... works extremely well, a great deal of effort has gone into making it user friendly as well as functional ... a very worthwhile package" *-CRASH*

"... an expert system ... incredibly easy to use" *-Popular Computing Weekly*

"... well produced ... a worthwhile addition to your toolkit" *-ZX Computing*

"... one of the most useful programs I have ever used. Makes 'xyz' look very childish and useless. A software company to keep an eye on" *-ZX Appeal (Canada)*

★ **COMPREHENSIVE**
+ 900 faults
+ 300 symptoms

★ **POWERFUL**
+ Discover faulty parts for every symptom
+ Display faults for every part
+ Recommended cure for every fault
+ Trace symptoms of any fault
+ Identify faults causing two symptoms
+ Mark items for easy reference
+ Eliminate faults checked OK
+ Review progress on summary screen
+ Call help if in difficulty
+ Save diagnosis in data file

CAR CURE will transform your micro into an **expert mechanic** to help you **find faults** on your car and then **determine the cure.**
Technical Colleges buy **CAR CURE** to help students **learn** about the faults cars suffer, as well as their symptoms and cures. Now you too can benefit!
Complete with comprehensive **5000-word User Guide** and packaged in a long-life plastic wallet. Export orders get American-English cross reference. All for less than many garages charge for an hour's work!

Cassette for **48K Spectrum** *(microdrive transfer instructions free, on request)* and **Amstrad CPC464** at only £9.95.
Disc for **BBCB** *(state 40 or 80 track)* at £11.95.

SIMTRON
PROGRAMS TO HELP YOU

DEPT ZX, 4 CLARENCE DRIVE, EAST GRINSTEAD, WEST SUSSEX RH19 4RZ Telephone (0342) 28188

XXX VIDEO

Telephone (day or night)
0373 - 830563
FOR DETAILS

Glossy list from: **XXX Video Club, 2 Streatham High Rd., London SW16**
(You must be over 18 to ask for details)

EDUCATIONAL SOFTWARE

SINCLAIR SPECTRUM 48K
MATHS (basic)
6-11 YRS
Addition, Subtraction, Tables, Multiplication, Division and Fractions. Timed exercises and pupil assessment. Various difficulty levels.

SPELLWELL
Randomly generated words of 5-10 letters. Different format option. Large letters. An excellent spellind aid for all ages.
£5.50 inc VAT and postage or £9.00 both cassettes.

EDUCATIONAL SOFTWARE
20 Beaufront Road, Camberley GU15 1NF. Tel: 0276 23819

BOOKS


Einstein

We use only 10% of our mental potentials

These are the words of Albert Einstein, the greatest physicist of recent times.
L Ron Hubbard's discoveries in the field of the mind prove today that Einstein was right.
In his book "DIANETICS: The Modern Science of Mental Health" L Ron Hubbard takes one more giant step in this direction.
He reveals how anyone can use his discoveries to isolate the exact barriers that have so far prevented people from using their mental potentials to the full.
More and more people from all walks of life use Dianetics technology today.
The many written reports on their success are the best proof of the quality of this book.
Find out for yourself. Order your copy of this remarkable book today.
Price £3.95 paperback £17.50 hardback. Make cheques and POs payable to S.H.F. Send to Church of Scientology, Saint Hill Foundation, FREEPOST (ZX1), East Grinstead, Sussex RH19 4ZA (no stamp required).

SCIENTIFIC

HALLEYS COMET in Planetarium setting with Orrery. A superb educational production for Spectrum 48K. See Halley's visit in the comfort of your own home. Produce star maps or planetary orbits for any date, today, yesterday or the future. An astronomical experience. Only £8.95 from Anima Scientific Computing, 33 Lilac Walk, Hebburn, Tyne & Wear.

UTILITIES

INTERFACE 007

The Ultimate device to transfer ANY program, even Hyper Loads to a Microdrive. (Wafa/Disc-drive versions available shortly). Simply load as normal, press a button on Interface 007 and the program freezes. Press a key & a special copy Saved to tape. Reload this by MERGE "" Enter GO TO 2 and program virtually jumps onto cartridge. Cartridges Load independent of Interface.

INTERFACE 007 . £29.95
(plus £1.50 Reg postage)
O/seas: Europe +£2 Others +£4

ZX-GUARANTEED (Dept ZX)
29 Chadderton Drive, Unsworth, Bury, Lancs. Tel: 061-766 5712

SOFTWARE

CONFUZION BY INCENTIVE £6.95

FREE SPECTRUM software loan when you join our library. Games educational/utilities available. Full details from 'Cygnet Software', 412 Fulbridge Road, Werrington, Peterborough, Cambs. PE4 6SE.

COLLECTORS! GEOLOGISTS! Identify those rocks and minerals with two quality programs and comprehensive instructions for the 48K Spectrum. Send £3.75 to D. Carmichael, 8 Sandown Close, Tunbridge Wells, Kent.

GAMES

ZX81 POOTER PUZZLER

4 Fast Programs containing 27 PUZZLES only £2·95

Traditional puzzle fun + SWITCH QUIZ! All games can be played with a joystick. Hours of fun for all ages. Only £2.95 post free. MONEY BACK IF NOT SATISFIED

POOTER GAMES
POOTER GAMES
24 PARSLOES AVENUE,
DAGENHAM RM9 5NX

RESERVE THIS SPACE RING 01-437 0699

FOR SALE

SINCLAIR 16K rampacks for ZX81. £6 including VAT & p&p. Halbar, Unit 1, Buny Walk, Bedford, Beds.

ACCESSORIES

SPECTRUM RGB OUTPUT MODULE
adapt electronics

- Connect a monitor to your Spectrum
- Super sharp text. Superb colour graphics •
- True TTL/Analogue outputs + comp syncs
- Monitor quality pictures – no dot crawl
- Suits RGB monitors & monitor/TVs
- Simply plugs into expansion port
- No separate power supply required
- New through port version. Allows connection of modems, joy-stick interfaces, etc.

Price £34.95 or £36.95 (through port) + £1 p&p many addons part & software SAE for details

20 Starling Close, Buckhurst Hill, Essex IG9 5TN
Tel. 01-504 2840

THE LEADER

ZX-GUARANTEED has established itself as the leading supplier of GENUINELY easy to use tape utility programs for handling ALL types of programs, even the Hyper jerky tone types. We also supply tapes to transfer your progs to Microdrive, etc. Send stamp for full list.

ZX-GUARANTEED (Dept ZX)
29 Chadderton Drive, Unsworth, Bury, Lancs. Tel: 061-766 5712

FOR HIRE

Hire & Exchange Software.
Exchange your software! Phone or send list of your software and ask for details. Software hire! No membership charge. P&P free. Hire tapes from £1.50. Phone for details.
T & P Software, 4 Maine Road, Newland, Drax, Selby, North Yorks. Tel: 0757 6187007.

ALARMS

BURGLAR ALARM Equipment. Please visit our 2,000 sq. ft. showrooms or write or phone for your free catalogue. CWAS Ltd., 100 Rooley Avenue, Bradford BD6 1DB. Telephone: (0274) 731532.

IK/16K ZX81 high quality software! Also A.Z.U.C. exclusive. ZX81 world wide club. For details send an SAE to: A.C.V., 1 Foxwell Square, Southfields, Northampton NN3 5AT, England.

UTILITIES

TAPE TO DRIVE TRANSFER OF RECENT PROGRAMS

NEW

SPECTRUM owners can even transfer many of the **LATEST** fast/jerky programs to their drives at **LOW COST** with our software and information sheets. No risk as our programs carry our **MONEY BACK GUARANTEE** (not updates).

Firstly you will need our MD1b (for m/drive) or WD1b (for Wafa or disc drive) programs that give you the tools to convert programs. Includes the ability to chop/split bytes in one go, bytes mover, "VAL" creator and REMkill, make visible etc. **FULL** manual with example transfers. **MD1b or WD1b cots on tape £6.99.** "MD1b manages more programs" — Your Spectrum.

To manage the latest programs you will also need our T7 program which INCLUDES the ability to ALTER fast/jerky loaders to "normal".

*Features Baud rate measurer *Jerky detector

T7 cost £6.50 on tape. On M/drive cartridge £9.99 (incl MT6, an extra M/drive utility).

NEW

LERM INFORMATION SHEETS

Each sheet gives you the DETAILED instructions (using our software) required to transfer to your drive 8 of the more POPULAR programs. They cost **£2 EACH plus SAE.** Up to sheet 5 available.

SPECIAL OFFER — TRANSFER PACK. Buy MD1b+T7 on M/Drive cartridge + INFORMATION SHEETS 1-3 for £15.50 (other drive owners get WD1b+T7 on tape + sheets for same price).

UPDATE SERVICE: Send old tape + large SAE to get a £2 REDUCTION. **OVERSEAS:** add £1 Europe, £2 others for each product. **SAE for enquiries.**

LERM, DEPT ZX, 10 BRUNSWICK GDNS, CORBY, NORTHANTS NN18 9ER.

ZX COMPUTING
CLASSIFIED ADVERTISEMENT — ORDER FORM

If you have something to sell now's your chance! Don't turn the page — turn to us!

Rates of charge: 40p per word per issue plus 15% VAT (minimum of 15 words). Please state classification and post to: **ZX COMPUTING, CLASSIFIED DEPARTMENT, 1 GOLDEN SQUARE, LONDON W1.**

					£5.25
					£7.00
					£8.75
					£10.50
					£12.25
					£14.00
					£15.75
					£17.50

Please place my advert in **ZX COMPUTING** for ...... issues commencing as soon as possible.

I am enclosing my Cheque/Postal Order/International Money Order for: (delete as necessary) £...... (Made payable to A.S.P. Ltd)

OR Debit my Access/Barclaycard (Delete as necessary)

All classified advertisements must be paid for in advance.

Please use BLOCK CAPITALS and include post codes.

Name (Mr/Mrs/Miss/Ms) ....
(delete accordingly)

Address ....

....

....

Signature .... **Date** ....

Daytime Tel. No. ....



THE COUNTDOWN HAS BEGUN
CRITICAL MASS
From DURELL
Commodore 64
Spectrum
R.R.P. £8.95
DURELL sales dept.,
Castle Lodge, Castle Green, Taunton TA1 4AB
Write now for details of Durell's Great New Competition also T shirts, Catalogue and Posters


SABOTEUR!
DURELL
software getting harder
SPECTRUM
&
AMSTRAD
R.R.P. £8.95
DURELL sales dept.,
Castle Lodge, Castle Green, Taunton TA1 4AB
Write now for details of Durell's
Great New Competition also
T-shirts, Catalogue
and Posters